INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A hp-like discontinuous Galerkin method for solving the 2D time-domain Maxwell's equations on non-conforming locally refined triangular meshes

Hassan Fahs — Stéphane Lanteri — Francesca Rapetti

N° 6162

February 2007

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--6162--FR+ENG

A *hp*-like discontinuous Galerkin method for solving the 2D time-domain Maxwell's equations on non-conforming locally refined triangular meshes

Hassan Fahs * , Stéphane Lanteri* , Francesca Rapetti †

Thème NUM — Systèmes numériques
Projet Caiman

Rapport de recherche n° 6162 — February 2007 — 41 pages

Abstract: This work is concerned with the design of a *hp*-like discontinuous Galerkin (DG) method for solving the 2D time-domain Maxwell's equations on non-conforming locally refined triangular meshes. The proposed DG method allows non-conforming meshes with arbitrary-level hanging nodes. This method combines a centered approximation for the evaluation of fluxes at the interface between neighboring elements of the mesh, with a leap-frog time integration scheme. It is an extension of the DG formulation recently studied in [18]. After reviewing the stability properties of the DG method introduced in [18], we present a new implementation of this method which allows local *h*-refinement and *p*-enrichment and which is based on a numerical quadrature formula for the computation of the flux matrices associated to non-conforming interfaces. Numerical experiments are presented which both validate the theoretical results of [18] and provide further insights regarding the practical performance of the resulting *hp*-like DG method, particulary when non-conforming locally refined meshes are employed.

Key-words: Maxwell's equations, discontinuous Galerkin method, non-conforming triangular meshes, energy conservation, *hp*-like method, numerical analysis.

* INRIA Sophia Antipolis, projet Caiman. Hassan.Fahs@sophia.inria.fr
† University of Nice/Sophia Antipolis, J. A. Dieudonné Mathematics Laboratory.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Une méthode Galerkin discontinu de type *hp* pour la résolution numérique des équations de Maxwell 2D en domaine temporel sur des maillages triangulaires localement raffinés non-conformes

Résumé : On présente dans ce rapport une méthode Galerkin discontinu (GD) de type *hp* pour la résolution numérique des équations de Maxwell 2D en domaine temporel sur des maillages triangulaires localement raffinés non-conformes. La méthode proposée autorise l'utilisation de maillages non-conformes présentant un nombre arbitraire de noeuds flottants. Cette méthode combine une approximation centrée pour l'évaluation des flux aux interfaces entre éléments voisins du maillage, à un schéma d'intégration en temps de type saute-mouton. Il s'agit d'une extension de la méthode GD récemment étudiée dans [18]. Après avoir rappeler les propriétés de stabilité de la méthode GD introduite dans [18], on présente une nouvelle implémentation de cette méthode qui autorise *h*-raffinement et *p*-enrichissement locaux et qui est basée sur une formule de quadrature pour calculer les matrices de flux associées aux interfaces non-conformes. Des expériences numériques sont présentées qui permettent d'une part, de valider les résultats théoriques de [18] et, d'autre part, d'avoir une première idée des performances de la méthode GD de type *hp*, en particulier dans le cas de maillages localement raffinés non-conformes.

Mots-clés : équations de Maxwell, méthode Galerkin discontinu, maillages triangulaires non-conformes, méthodes *hp*, stabilité, analyse numérique.

1 Introduction

It is well known that the difficulties linked to the numerical resolution of the time-domain Maxwell's equations find their roots not in the equations themselves, but rather in the data of the problem, i.e. electric size, geometric complexity, inhomogeneous discontinuous materials, and broadband complex signal types. Applications with such characteristics can be found throughout the applied sciences and engineering, e.g. the design and analysis of antennas [6] and radars [43], the design of emerging technologies such as high-speed electronics and integrated optics, optical communication and a variety of military and civilian applications [44]-[41]. Other challenging applications are addressing societal questions such as the potential adverse effects of electromagnetic waves emitted by mobile phones [47]. These are all problems characterized by being very large in terms of a characteristic wavelength, geometrically extremely complex, often composed of a heterogeneous collection of different materials and all requiring a high fidelity solution with a rigorous control of the numerical errors. Even for linear problems such conditions forces one to look beyond standard computational techniques and seek new computational frameworks enabling the accurate, efficient, and robust modeling of wave-phenomena over long times in settings of a realistic geometric complexity.

The simplicity and reasonable accuracy of the classical finite difference time-domain (FDTD) method [57] has propelled this method to become the method of choice among engineers and scientists solving Maxwell's equations in the time-domain. In particular the last decade has seen an increase in applications and further developments of this method, many driven from the very influential texts by Taflove [50]. It is also, however, clear that the FDTD methods have severe limitations: its inherent second order accuracy severely limits its ability to correctly represent wave motion over long distances unless the grid is prohibitively fine. Furthermore, the simplicity of the method, often recognized as its strength, also becomes its most severe restriction by prohibiting the accurate representation of problems in complex geometries. In recent years, a number of efforts have been aimed at addressing the shortcomings of the classical FDTD schemes, e.g. embedding schemes to overcome staircasing [17]-[56], high-order finite difference schemes [50]-[21], body conforming nonorthogonal [20]-[25] or non-conforming orthogonal FDTD methods [58]-[13] or contour path methods [31]. Other techniques and improvements are discussed in [50]. Most of these methods, however, have not really penetrated into main stream user community, partly due to their complicated nature and partly because these methods themselves often introduce other complications.

For dealing with complex geometries, one can resort to use finite volume time-domain (FVTD) methods [9]-[40] or finite element time-domain (FETD) methods [35]. These methods are, in their classical form, second order accurate as the FDTD method. The existing FVTD methods are mostly based on upwind or partially upwind fluxes, along with multi-step low-storage Runge-Kutta time-schemes, and less often on centered fluxes along with a leap-frog scheme, a non-diffusive combination which was first proposed by Remaki [46]. One can notice here that, although Runge-Kutta schemes are robust and stable, no stability proof is available for these kind of schemes on arbitrary meshes [34] (stability is proved for the centered leap-frog FVTD methods on either cartesian grids [46] or unstructured meshes [40]). Furthermore, instability phenomenon have been observed in the case of non-conforming orthogonal meshes (see chapter 2 in [5]).

The FETD methods can handle unstructured meshes and complex geometries however, the development of such methods for solving Maxwell's equations, especially those with high-order accuracy, has been relatively slow. A primary reason is the appearance of spurious, non-physical solutions when a straightforward nodal continuous Galerkin finite element scheme is used to approximate

the Maxwell curl-curl equations. The source of these problems have several interpretations among which a poor representation of the large null space of the operator [2] and the generation of solutions that violate the divergence conditions, which are typically not imposed directly [39]. A discussion of these issues can be found in [49]. Bossavit made the fundamental observation that the use of special curl-conforming elements [37]-[36] would overcome the problem of spurious modes by mimicking properties of vector algebra [2]-[3]. Although very successful, such formulations are not entirely void of problems: the algebraic problems are larger than for nodal elements and the conformity requirements of the continuous Galerkin formulation makes adaptivity complex.

Demkowicz and co-workers pioneered the application of *hp*-adaptive finite element techniques to electromagnetics [14]-[15]-[52]. In their early two-dimensional work, they proposed an arbitrary order triangular element and performed calculations with uniform polynomial order. The convergence analysis was presented in [52]. Then, they described in [14] and [38] a complete finite element package which allowed for the use of hybrid quadrilateral/triangular meshes with simultaneous h-refinement and p-enrichment. A series of model problems were solved, without adaptivity [15]-[52], to demonstrate the effectiveness of p-enrichment. Computations undertaken with h-adaptivity were also reported [14]-[38]. More recently, Rachowicz and Demkowicz [42] have developed a three-dimensional *hp*-finite element package based upon hexahedral elements, see also [7]-[43]. Their approach allows for meshes with hanging nodes and for anisotropic refinement. They describe a data structure which allows such a refinement, although they do not currently have a strategy for the correct selection of the combination of h-refinement and p-enrichment. Furthermore, curved elements are allowed through the definition of a reference element and a geometric mapping. Numerical examples illustrate the effectiveness of the approach and these include scattering by rectangular waveguides, scattering by a radiating dipole, the absorption of electromagnetic waves by the human head and scattering of an electromagnetic wave by a generic aircraft. One can note here that to our knowledge, all existing *hp*-adaptive FETD software only allow one level hanging nodes, excepting the recent work of Solín *et al.* [48] which proposes a new automatic adaptive strategy for *hp*-FEM based on arbitrary-level hanging nodes, and which can be applied to standard low-order and spectral finite element methods.

In an attempt to offer an alternative to the classical finite element formulation based on edge elements, we consider here discontinuous Galerkin formulations [10] based on high-order nodal elements for solving the first-order time-domain Maxwell's equations. Discontinuous Galerkin time-domain (DGTD) methods based on discontinuous finite element spaces, easily handle elements of various types and shapes, irregular non-conforming meshes [18], and even locally varying polynomial order, and hence offer great flexibility in the mesh design, but they also lead to (block-) diagonal mass matrices and therefore yield fully explicit, inherently parallel methods when coupled with explicit time stepping [1]. Indeed, the mass matrix arising from a DGTD discretization is always block-diagonal, with block-size equal to the number of degrees of freedom per element; hence, it can be inverted at very low computational cost. In fact, for constant material coefficients, the mass matrix is diagonal for a judicious choice of (locally orthogonal) shape functions [45]. Here continuity is weakly enforced across mesh interfaces by adding suitable bilinear forms (so-called numerical fluxes) to the standard variational formulations.

The DGTD methods were originally devised to solve scalar first-order hyperbolic problems but were later generalized to apply to first- and second-order hyperbolic systems included Maxwell's equations and the acoustic wave equation [51]. Such formulations have been successful in solving Maxwell's equations in the time-domain, which schemes based on upwind fluxes [32]-[54]-[22]-[23]. In [12], for instance, Cockburn and Shu use a discontinuous Galerkin formulation in space combined with a Runge-Kutta scheme in time to discretize hyperbolic conservation laws. In [32]-[33], Kopriva

et al. developed a discontinuous Galerkin method which combines high-order spectral elements with a fourth order low-storage Runge-Kutta scheme, using conforming and non-conforming meshes. Warburton [54], Hesthaven and Warburton [22] used a similar approach for their Runge-Kutta discontinuous Galerkin (RKDG) method, which combines high-order spatial accuracy with a fourth order low-storage Runge-Kutta scheme. While successful their scheme does not conserve energy due to upwinding. Recently, Chen *et al.* developed a high-order RKDG method for Maxwell's equations in first-order hyperbolic form, which achieves high-order convergence both in space and time by using a strong stability preserving (low storage) SSP-RK scheme [8]. By using locally divergence-free polynomials Cockburn *et al.* developed a locally divergence-free DGTD method for the first-order Maxwell's equation [11].

Recently, Piperno *et al.* developed a DGTD method in three dimensions using centered fluxes combined with a leap-frog scheme [19]. Their scheme conserves a discrete analog of the electromagnetic energy and preserves the divergence relations (in a certain sense) in the absence of sources, but the convergence rate of this scheme remains sub-optimal. Canouet *et al.* developed a new DGTD scheme using the same approach (centered fluxes and leap-frog scheme) on locally refined cartesian meshes [6]. Finally, for the frequency-domain Maxwell's equations, discontinuous Galerkin methods on simplicial meshes have been studied by Hesthaven and Warburton [24] and Houston and co-workers [29]-[4], while orthogonal non-conforming meshes are considered in [27]-[28], using 1- and 3-irregular meshes.

The rest of this report is organized as follows. In section 2 we recall the basic features of our discontinuous Galerkin time-domain formulation for solving the first order Maxwell's equations in the time domain, based on totally centered numerical fluxes and a leap-frog time-integration scheme. Section 3 is concerned with the treatment of non-conforming meshes and in particular, the computation of the flux matrices associated to non-conforming interfaces. In section 4, we present numerical results for non-conforming calculations based on h-refinement and p-enrichment . The hp-like DGTD method is presented in section 5 and a numerical convergence analysis is conducted in section 6. Finally, conclusions and future works are summarized in section 7.

2 The discontinuous Galerkin scheme

Let Ω be a bounded Lipschitz and simply connected domain in $\mathbb{R}^2$, with $\vec{n}$ denoting the unit normal vector pointing outward to boundary $\Gamma = \partial\Omega$, which we assume to be connected. We shall consider the solution of the two-dimensional Maxwell's equations in the TM polarization on the domain Ω. We shall assume that the field components as well as the material parameters ϵ and μ do not depend on the z coordinate of a point. The equations are :

$$\begin{cases} \epsilon \dfrac{\partial E_z}{\partial t} - \dfrac{\partial H_y}{\partial x} + \dfrac{\partial H_x}{\partial y} = 0, \\ \mu \dfrac{\partial H_x}{\partial t} + \dfrac{\partial E_z}{\partial y} = 0, \\ \mu \dfrac{\partial H_y}{\partial t} - \dfrac{\partial E_z}{\partial x} = 0, \end{cases} \tag{2.1}$$

where the unknowns are E_z and $\mathbf{H} = (H_x, H_y)$, the electric and magnetic fields, respectively. The coefficients ϵ and μ are the electric permittivity and the magnetic permeability of the medium, respectively; they are assumed to be piecewise constant. The boundary Γ is assumed to be a perfect electric conductor and the corresponding boundary conditions are :

$$E_z = 0 \quad , \quad \vec{n}.\mathbf{H} = 0,$$

indicating that the total tangential electric and normal magnetic field components must vanish.

Throughout this work, we shall consider shape-regular and affine meshes $\mathcal{T}_h$ that partition the domain Ω into $N = \text{Card}(\mathcal{T}_h)$ disjoint triangles T_i with boundaries ∂T_i such that $\overline{\Omega} = \cup\{\overline{T}_i, T_i \in \mathcal{T}_h\}$. The diameter of element T_i is denoted by h_i, and the mesh size h is given by $h = \max_{T_i \in \mathcal{T}_h} h_i$. Within this construction we admit meshes with possibly hanging nodes i.e. by allowing irregular (or non-conforming) meshes where element vertices can lie in the interior of edges of other elements (see Fig. 2.1 and subsection 2.1). Each triangle T_i is assumed to be the image, under a smooth bijective (diffeomorphic) mapping τ_i, of a fixed reference triangle $\hat{\mathrm{T}} = \{\hat{x}, \hat{y} |\ \hat{x}, \hat{y} \geq 0; \hat{x} + \hat{y} \leq 1\}$. Assuming that T_i is a straight sided triangle defined through the coordinates of the three vertices $\mathbf{v}_1^i$, $\mathbf{v}_2^i$ and $\mathbf{v}_3^i$ (see Fig. 2.2), the correspondence between the two triangles $\hat{\mathrm{T}}$ and T_i is established through the use of the barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$. We recall that any point $\mathbf{x}^i \in T_i$ can be expressed as a convex combination of the vertices of T_i and the mapping is simply given by $\tau_i : (\hat{x}, \hat{y}) \in \hat{\mathrm{T}} \rightarrow \mathbf{x}^i$, such that:

$$\mathbf{x}^i(\hat{x}, \hat{y}) = \lambda_1 \mathbf{v}_1^i + \lambda_2 \mathbf{v}_2^i + \lambda_3 \mathbf{v}_3^i$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $0 \leq (\lambda_1, \lambda_2, \lambda_3) \leq 1$ with $\lambda_1 = 1 - \hat{x} - \hat{y},\ \lambda_2 = \hat{x}$ and $\lambda_3 = \hat{y}$.

For each triangle T_i, we denote by $|T_i|$ its surface, and ϵ_i and μ_i are respectively the local constant electric permittivity and magnetic permeability, and c_i the local speed of propagation. For two distinct triangles T_i and T_k in $\mathcal{T}_h$, the intersection $T_i \cap T_k$ is an (oriented) edge a_{ik} which we will call interface, with oriented normal vector $\vec{n}_{ik}$ and with unitary one $\vec{\tilde{n}}_{ik}$. For the boundary faces, the index k corresponds to a fictitious element outside the domain. Finally, we denote by $\mathcal{V}_i$ the set of indices of the elements neighboring T_i. We also define the perimeter in x and y direction by $P_i^{\mathbf{x}} = \sum_{k \in \mathcal{V}_i} |n_{ik\mathbf{x}}|$, where $\mathbf{x} = \{x, y\}$.

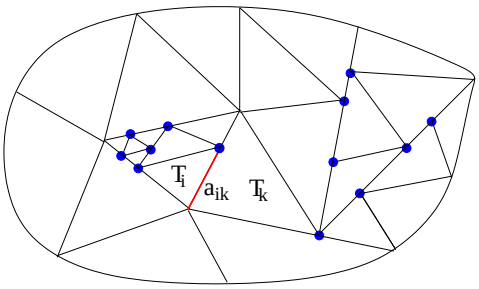


Figure 2.1: Irregular mesh with blue hanging nodes.

To each $T_i \in \mathcal{T}_h$ we assign an integer $p_i \geq 0$ and we collect the p_i in the vector $p = \{p_i : T_i \in \mathcal{T}_h\}$. For a given partition $\mathcal{T}_h$ of Ω and vector p, we seek approximate solutions to (2.1) in the finite dimensional subspace $V_p(\mathcal{T}_h)$ defined as follows:

$$V_p(\mathcal{T}_h) := \{v \in L^2(\Omega) : v_{|T_i} \in \mathbb{P}^{p_i}(T_i)\ ,\ \forall T_i \in \mathcal{T}_h\}, \tag{2.2}$$

where $\mathbb{P}^{p_i}(T_i)$ denotes the space of polynomials of total degree at most p_i on the element T_i. The space $V_p(\mathcal{T}_h)$ has the dimension $d_i = \frac{(p_i+1)(p_i+2)}{2}$ and d_i denotes the local number of degrees of

freedom inside the finite element T_i. Note that the polynomial degree, p_i, may vary from element to element in the mesh and that a function $v_h^p \in V_p(\mathcal{T}_h)$ is discontinuous across element interfaces. By non-conforming interface we mean an interface a_{ik} which has at least one of its two vertices in a hanging node or/and $p_{i|_{a_{ik}}} \neq p_{k|_{a_{ik}}}$.

For any field $\mathbf{X} \in \{E_z, H_x, H_y\}$, we denote by $\mathbf{X}_i$ the L^2-projection on the linear space $\mathsf{Span}\{\phi_{ij},\ 1 \leq j \leq d_i\}$ spanned by functions defined on T_i, and where $\{\phi_{ij}\}_{j=1,\ldots,d_i} \in \mathbb{P}^{p_i}(T_i)$ is a family of linear independent fields. In each triangle T_i we construct a polynomial representation of the magnetic and electric fields and for simplicity and efficiency reasons we adopt a Lagrangian interpolation approach based on a set of nodes defined on the master (reference) triangle $\hat{\mathrm{T}}$ (see [16], pp. 106-120). Using this notation, we have the following global discontinuous representation of the field:

$$\mathbf{X} \simeq \sum_i \mathbf{X}_i(t,x,y) = \sum_i \sum_{j=1}^{d_i} X_{ij}(t)\varphi_{ij}(x,y), \tag{2.3}$$

where X_{ij} is the jth degree of freedom of the field $\mathbf{X}_i$. We denote by X_i the column vector $(X_{ij})_{1\leq j\leq d_i}$. The approximation field $\mathbf{X}_h \in \{(E_z)_h, (H_x)_h, (H_y)_h\}$, defined by $(\mathbf{X}_{h|T_i} = \mathbf{X}_i, \forall i)$ is allowed to be discontinuous across element boundaries and, for such a discontinuous field $\mathbf{X}_h$, we define its average $\{\mathbf{X}_h\}_{ik}$ on any internal face a_{ik}, as:

$$\{\mathbf{X}_h\}_{ik} = \frac{\mathbf{X}_{i|a_{ik}} + \mathbf{X}_{k|a_{ik}}}{2}. \tag{2.4}$$

Note that for any internal face a_{ik}, $\{\mathbf{X}_h\}_{ik} = \{\mathbf{X}_h\}_{ki}$. For any integral over ∂T_i, a specific treatment must be introduced since the approximate fields are discontinuous through element faces. We choose to use a centered approximation:

$$\forall i, \forall k \in \mathcal{V}_i, \mathbf{X}_{|a_{ik}} \simeq \{\mathbf{X}_h\}_{ik}. \tag{2.5}$$

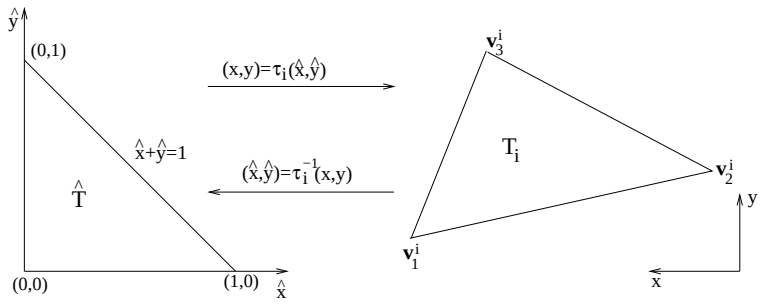


Figure 2.2: Mapping between the physical triangle T_i and the master triangle $\hat{\mathrm{T}}$.

For what concerns time discretization, we propose to use a leap-frog time integration scheme which has both the advantage to be explicit and free of time dissipation. In the sequel, superscripts refer to time stations and Δt is the fixed time-step. The unknowns related to the electric field are approximated at integer time stations $t^n = n\Delta t$ and are denoted by E_{zi}^n. The unknowns related to the magnetic field are approximated at half-integer time stations $t^{n+\frac{1}{2}} = (n+\frac{1}{2})\Delta t$ and are denoted by $H_{xi}^{n+\frac{1}{2}}$ and $H_{yi}^{n+\frac{1}{2}}$.

The perfectly electrically conducting (metallic) boundary condition on Γ is treated in a weak sense by defining some values for the fields E_z, H_x and H_y in the fictitious neighboring element T_k. The

treatment of boundary conditions is weak in the sense that the traces on $a_{ik} \in \partial\Omega$ of fictitious fields $E_{zk}^n, H_{xk}^{n+\frac{1}{2}}$ and $H_{yk}^{n+\frac{1}{2}}$ are used for the computation of numerical fluxes for the boundary element T_i. More precisely, for a metallic boundary face a_{ik}, the fictitious values are chosen as :

$$\forall (x,y) \in a_{ik} : \left\{ \begin{array}{l} H_{xk}^{n+\frac{1}{2}}(x,y) = H_{xi}^{n+\frac{1}{2}}(x,y), \quad H_{yk}^{n+\frac{1}{2}}(x,y) = H_{yi}^{n+\frac{1}{2}}(x,y), \\ E_{zk}^n(x,y) = -E_{zi}^n(x,y). \end{array} \right. \tag{2.6}$$

The discontinuous Galerkin $\mathbb{P}_{p_i}$-DGTD scheme can be written in the following matrix form:

$$\left\{ \begin{array}{rcl} \epsilon_i M_i \left(\dfrac{E_{zi}^{n+1} - E_{zi}^n}{\Delta t} \right) & = & -K_i^x H_{yi}^{n+\frac{1}{2}} + K_i^y H_{xi}^{n+\frac{1}{2}} + \displaystyle\sum_{k \in \mathcal{V}_i} \left(G_{xik}^{n+\frac{1}{2}} - G_{yik}^{n+\frac{1}{2}} \right), \\ \mu_i M_i \left(\dfrac{H_{xi}^{n+\frac{1}{2}} - H_{xi}^{n-\frac{1}{2}}}{\Delta t} \right) & = & K_i^y E_{zi}^n - \displaystyle\sum_{k \in \mathcal{V}_i} F_{yik}^n, \\ \mu_i M_i \left(\dfrac{H_{yi}^{n+\frac{1}{2}} - H_{yi}^{n-\frac{1}{2}}}{\Delta t} \right) & = & -K_i^x E_{zi}^n + \displaystyle\sum_{k \in \mathcal{V}_i} F_{xik}^n, \end{array} \right. \tag{2.7}$$

where the vector quantities $F_{xik}^n, F_{yik}^n, G_{xik}^{n+\frac{1}{2}}$ and $G_{yik}^{n+\frac{1}{2}}$ are defined as:

$$\left\{ \begin{array}{ccc} F_{xik}^n = S_{ik}^x E_{zk}^n & , & F_{yik}^n = S_{ik}^y E_{zk}^n, \\ G_{xik}^{n+\frac{1}{2}} = S_{ik}^x H_{yk}^{n+\frac{1}{2}} & , & G_{yik}^{n+\frac{1}{2}} = S_{ik}^y H_{xk}^{n+\frac{1}{2}}, \end{array} \right. \tag{2.8}$$

and the positive definite symmetric mass matrix M_i and the skew-symmetric stiffness matrix $K_i^{\mathbf{x}}, \mathbf{x} \in \{x, y\}$ (all of size $d_i \times d_i$) are given by:

$$\left\{ \begin{array}{l} (M_i)_{jl} = \displaystyle\int_{T_i} \varphi_{ij} \varphi_{il}, \\ (K_i^{\mathbf{x}})_{jl} = \dfrac{1}{2} \displaystyle\int_{T_i} \left(\frac{\partial \varphi_{ij}}{\partial \mathbf{x}} \varphi_{il} - \varphi_{ij} \frac{\partial \varphi_{il}}{\partial \mathbf{x}} \right). \end{array} \right. \tag{2.9}$$

For any interface a_{ik}, the $d_i \times d_k$ rectangular interface matrix $S_{ik}^{\mathbf{x}}, \mathbf{x} \in \{x, y\}$ is given by:

$$(S_{ik}^{\mathbf{x}})_{jl} = \frac{1}{2} \widetilde{n}_{ik\mathbf{x}} \int_{a_{ik}} \varphi_{ij} \varphi_{kl}. \tag{2.10}$$

For the interface matrix S_{ik}, we have the following properties:

- if a_{ik} is an internal face of the mesh then ${}^t S_{ik}^{\mathbf{x}} = -S_{ki}^{\mathbf{x}}$,
- if a_{ik} is a metallic boundary face of the mesh then ${}^t S_{ik}^{\mathbf{x}} = S_{ik}^{\mathbf{x}}$.

In [18] a preliminary implementation of the discontinuous Galerkin schemes (2.7)-(2.8)-(2.9)-(2.10) has been used where an approximated (simplified) calculation of the interface matrix (2.10) was adopted. In the present work, for non-conforming interfaces, we calculate the interface matrix (2.10) by using a numerical quadrature formula. We will motivate this choice in section 3.

2.1 Hanging nodes and irregular meshes

We review here different kinds of local mesh refinement strategies which can be used for the discretization of complex geometries or deformable bodies. Moreover, refining locally the mesh is expected to reduce the computation in time and memory. When working with regular meshes (where two elements either share a common vertex, common edge, or their intersection is empty), adaptivity is often done using the so-called *red-green* refinement strategy [26]-[30]. This technique first subdivides desired elements into geometrically convenient subelements with hanging nodes and then eliminates the hanging nodes by forcing refinement of additional elements, as illustrated on Fig. 2.3.

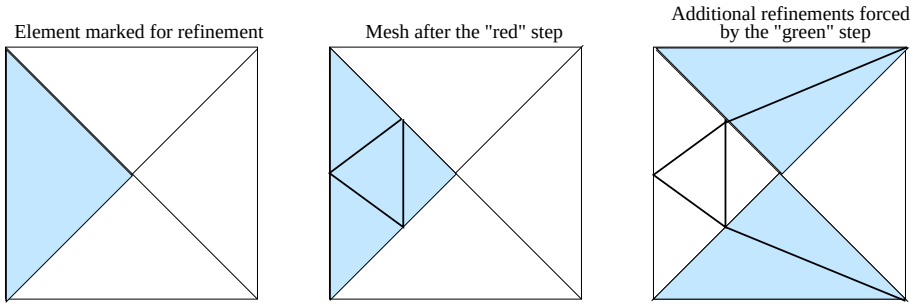


Figure 2.3: Red-green refinement of a triangular mesh.

This approach preserves the regularity of the mesh but at the same time it introduces elements with sharp angles. Often, this happens when repeated refinements occur in some part of the domain, for instance in a boundary layer or near a geometrical singularity. The *green* refinements can be avoided by introducing *hanging* nodes, i.e. by allowing irregular (or non-conforming) meshes where element vertices can lie in the interior of edges of other elements. To ease the computer implementation, most discontinuous Galerkin finite element code working with hanging nodes limit the maximum difference of refinement levels on adjacent elements to 1 (1-irregularity rule, see among others [28]-[45]).

In the following, we only consider the *red* refinements and by κ-irregularity rule (or κ-level hanging nodes, or κ-irregular mesh) we refer to a restriction where the maximum difference of refinement levels of adjacent elements is κ. In this context, $\kappa = 0$ corresponds to adaptivity with regular meshes and $\kappa = \infty$ to adaptivity with arbitrary-level hanging nodes. As illustrated on Fig. 2.4, even the 1-irregularity rule does not avoid all forced refinements.
The amount of forced refinements in the mesh depends strongly on the level of hanging nodes. Fig. 2.5 shows a more general irregular mesh, where hanging nodes are not uniformly spaced along a given edge.

2.2 Stability of the discontinuous Galerkin scheme

We define the following discrete version of the electromagnetic energy:

$$\mathcal{E}^n = \frac{1}{2}\sum_{i=1}^{N}\Big(\epsilon_i{}^t E_{zi}^n M_i E_{zi}^n \;+\; \mu_i{}^t H_{xi}^{n-\frac{1}{2}} M_i H_{xi}^{n+\frac{1}{2}} \;+\; \mu_i{}^t H_{yi}^{n-\frac{1}{2}} M_i H_{yi}^{n+\frac{1}{2}}\Big). \tag{2.11}$$

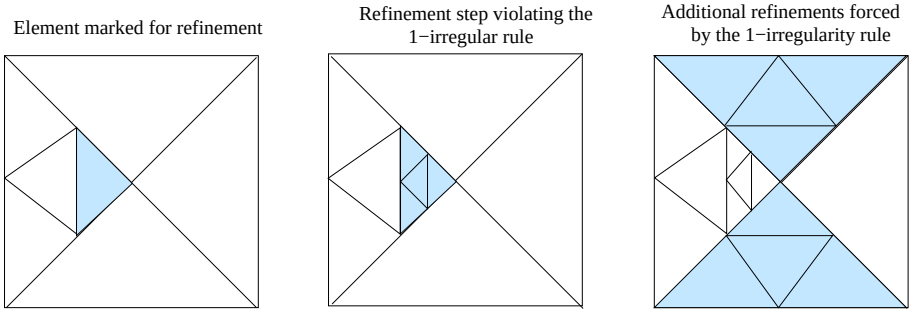


Figure 2.4: Red refinement with 1-irregularity rule or 1-irregular mesh.

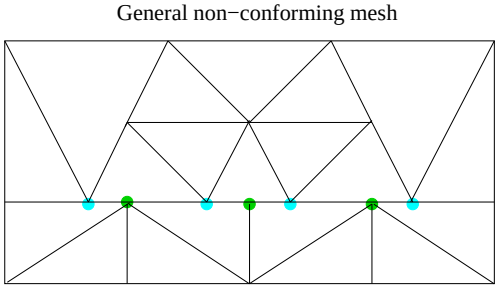


Figure 2.5: General irregular (non-conforming) mesh.

The stability of the discontinuous Galerkin $\mathbb{P}_{p_i}$-DGTD scheme (2.7)-(2.8)-(2.9)-(2.10) has been studied in [18], for non-conforming meshes and any local polynomial degree p_i on T_i. The scheme is conditionally stable under a CFL-like condition on the time-step Δt which depends on p_i and on the level of hanging nodes κ. The basic idea is to prove that the energy (2.11) is a quadratic form who plays the role of a Lyapunov function of the whole set of numerical unknowns. The CFL conditions are defined as follows (for more details we refer the reader to [18]) :

$$\left\{ \begin{array}{ll} \forall \text{ internal face } a_{ik} \ (k \in \mathcal{V}_i), & \Delta t \leq \min(\Delta t_1, \Delta t_2), \\ \forall \text{ metallic face } a_{ik} \ (k \in \mathcal{V}_i), & c_i \Delta t (2\alpha_i + \beta_{ik}) \leq 4 \min(\dfrac{|T_i|}{P_i^x}, \dfrac{|T_i|}{P_i^y}). \end{array} \right. \tag{2.12}$$

where the time steps Δt_1 and Δt_2 are such that:

$$\begin{aligned} c_i \Delta t_1 (2\alpha_i + \beta_{ik}) &\leq 4 \min\Big(\frac{|T_i|}{P_i^x}, \frac{|T_i|}{P_i^y}\Big), \\ c_k \Delta t_2 (2\alpha_k + \beta_{ki}) &\leq 4 \min\Big(\frac{|T_k|}{P_k^x}, \frac{|T_k|}{P_k^y}\Big). \end{aligned} \tag{2.13}$$

The constants α_i and β_{ik} (α_k and β_{ki} respectively) verify some inverse inequalities [55]-[53] on T_i and a_{ik} (on T_k and a_{ki} respectively) :

$$\left\{ \begin{array}{l} \forall \xi_i \in \mathsf{Span}\{\varphi_{ij}, 1 \leq j \leq d_i\} \ , \ \mathbf{x} \in \{x, y\} \\ \left\| \dfrac{\partial \xi_i}{\partial \mathbf{x}} \right\|_{T_i} \leq \dfrac{\alpha_i P_i^{\mathbf{x}}}{|T_i|} \|\xi_i\|_{T_i}, \quad \|\xi_i\|_{a_{ik}}^2 \leq \beta_{ik} \dfrac{\|\vec{n}_{ik}\|}{|T_i|} \|\xi_i\|_{T_i}^2. \end{array} \right.$$

Moreover, these constants depend on the local polynomial degree p_i (see [18] for more details). In [18], we have remarked that the analytical CFL condition (2.12) is restrictive, especially in the case of irregular meshes, when the local level of refinement increases. This comes from the fact that the calculation of the constant β_{ik} depends on some optimization problem on $a_{ik}, \forall i, \forall k \in \mathcal{V}_i$, defined as follows : find a positive constant β_{ik} such that, $A_i \preceq \beta_{ik} M_i$, where M_i is the mass matrix in the triangle T_i defined in (2.9) and A_i is the symmetric matrix defined by $(A_i)_{jj'} = \int_{a_{ik}} \varphi_{ij}\varphi_{ij'}$. If a_{ik} is a non-conforming interface, then the β_{ik} can take large values according to the position and length of a_{ik} (see subsection 3.2 in [18]).

3 Calculation of the interface matrix

The aim of this section is to present two methods to calculate the interface matrix (2.10) when the a_{ik} is a non-conforming face. We restrict ourselves to an affine family of finite elements and the results presented here correspond to the 2D case. Part of these results can be easily generalized to the 3D case.

3.1 The projection and quadrature based approaches

We denote by S_F the interface matrix whose coefficients are defined in the following way :

$$(S_F)_{ij} = \int_F \phi_i \psi_j ds, \forall i \in \{1, \ldots, n\},\ \forall j \in \{1, \ldots, m\}. \tag{3.14}$$

Here, $F = \partial T_1 \cap \partial T_2$ refers to a non-conforming face of the mesh. The functions $(\phi_i)_{i=1}^n$ and $(\psi_j)_{j=1}^m$ are the restriction to the face F of basis functions defined on the elements T_1 and T_2 and not identically equal to zero on F (see Fig. 3.6 (a) and (b)). They span spaces of polynomial functions of some degree defined on F.

Let $(\gamma_k)_{k=1}^n$ be a second basis of the space spanned by $(\phi_i)_{i=1}^n$, and $(\zeta_l)_{l=1}^m$ a second basis of the space spanned by $(\psi_j)_{j=1}^m$. Let P^1 and P^2 respectively be the change of coordinate matrices from bases $(\phi_i)_{i=1}^n$ and $(\psi_j)_{j=1}^m$ to bases $(\gamma_k)_{k=1}^n$ and $(\zeta_l)_{l=1}^m$. More precisely:

$$\begin{cases} \phi_i &= \sum_{k=1}^{n} P^1_{ik} \gamma_k,\ \forall i \in \{1, \ldots, n\}, \\ \psi_j &= \sum_{l=1}^{m} P^2_{jl} \zeta_l,\ \forall j \in \{1, \ldots, m\}. \end{cases}$$

We have the following lemma.

Lemma 1 *Let $A_{F'}$ be the matrix defined by:*

$$(A_{F'})_{ij} = \int_{F'} \gamma_i \zeta_j ds,\ \forall i \in \{1, \ldots, n\},\ \forall j \in \{1, \ldots, m\}.$$

where F' denotes a conforming interface (the same that F but in a conforming situation, see Fig. 3.6 (c) and (d)). Then:

$$S_F = P^1 A_{F'} (P^2)^t. \tag{3.15}$$

The formula (3.15) can be used to evaluate S_F and this approach will be referred in the sequel as *projection approach*. The matrices S_F and $A_{F'}$ are both of size $n \times m$ and the matrices P^1 and P^2 are of size $n \times n$ and $m \times m$ respectively.

PROOF : The calculation steps are as follows:

$$\begin{aligned}
(S_F)_{ij} &= \int_F \left(\sum_{k=1}^{n} P^1_{ik} \gamma_k \right) \left(\sum_{l=1}^{m} P^2_{jl} \zeta_l \right) ds \\
&= \sum_{k=1}^{n} P^1_{ik} \left(\sum_{l=1}^{m} \left(\int_{F \equiv F'} \gamma_k \zeta_l ds \right) P^2_{jl} \right) \\
&= \sum_{k=1}^{n} P^1_{ik} \left(\sum_{l=1}^{m} (A_{F'})_{kl} P^2_{jl} \right) \\
&= \sum_{k=1}^{n} P^1_{ik} \left(\sum_{l=1}^{m} (A_{F'})_{kl} ((P^2)^t)_{lj} \right) \\
&= \sum_{k=1}^{n} P^1_{ik} \left(A_{F'} (P^2)^t \right)_{kj} \\
&= (P^1 A_{F'} (P^2)^t)_{ij}.
\end{aligned}$$

□

On the other hand, the matrix S_F can be evaluated using a numerical quadrature formula based on the zeros of Legendre polynomials, i.e., the nodes of the Gauss-Legendre quadrature formula. We denote by N_G the maximal number of nodes and weights necessary to obtain an exact integration of polynomials of degree $\leq 2N_G - 1$. A matrix representation of the operations to be carried out in this case is:

$$S_F \simeq Q^1 B (Q^2)^t, \tag{3.16}$$

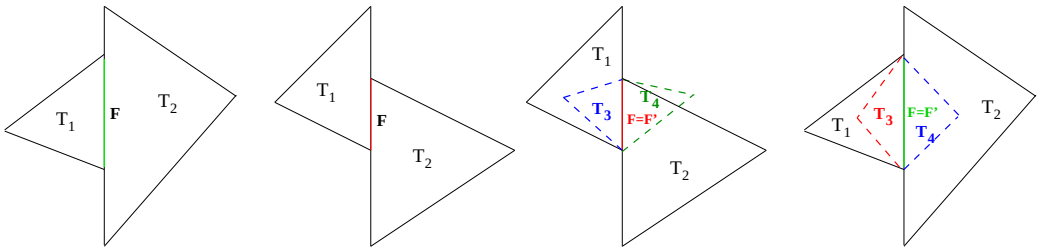
where the matrices Q^1 and Q^2 are respectively of size $n \times N_G$ and $m \times N_G$, while the matrix B is diagonal of size $N_G \times N_G$. The coefficients of these matrices are defined by:

$$\left\{ \begin{array}{lcll}
(Q^1)_{ik} & = & \phi_i(x_k) & , \ \forall 1 \leq i \leq n, \ \forall 1 \leq k \leq N_G, \\
(Q^2)_{jk} & = & \psi_j(x_k) & , \ \forall 1 \leq j \leq m, \ \forall 1 \leq k \leq N_G, \\
(B)_{kk} & = & \omega_k & \text{and} \quad (B)_{rk} = 0, \ \forall r \neq k, \ \forall 1 \leq k \leq N_G.
\end{array} \right.$$

The real numbers x_k and ω_k are respectively the nodes and weights of Gauss-Legendre. In the sequel, the evaluation of S_F using the formula (3.16) will be referred as *quadrature approach*.

The reason of the choice of Gauss-Legendre is that the weight function of the Legendre polynomial is equal to one. Nevertheless, we can use Gauss-Lobatto quadrature, where both endpoints of the integration domain are quadrature nodes and the problem in this case is to choose the weights ω_k and the remaining nodes x_k to maximize the degree of exactness of the quadrature rule. Note that, for Gauss-Lobatto we only need $N_G - 2$ points and weights (two of them are the endpoints of the integration domain) to obtain an exact integration of polynomials of degree $\leq 2N_G - 3$. Without entering into details, let us note however, that to reduce the storage cost, only nodes and

the associated weights in the interval $[0;0.5]$ are stored $\big(\frac{N_G+1}{2}$ for Gauss-Legendre and $\frac{N_G-1}{2}$ for Gauss-Lobatto$\big)$ and we can find the remaining values in $[0.5;1]$ by symmetry arguments.



(a) (b) (c) (d)

Figure 3.6: Fig. (a) and (b) show two possible configurations for a non-conforming mesh. On Fig. (c) and (d), we remark that the interface F is a non-conforming interface for the triangles T_1 and T_2, on the other hand, it is a conforming interface for the triangles T_3 and T_4 and we call it F'.

3.2 Comparison between the two approaches

The objective here is to compare the projection approach (3.15) with the quadrature approach (3.16) from the point of view of the computational complexity. In order to do so, we summarize in Tab. 1 the number of arithmetic operations required to calculate the matrix S_F with the two approaches.

	Projection approach	Quadrature approach
Multiplications	$nm(n+m)$	$nN_G(m+1)$
Additions	$nm(n+m-2)$	$nm(N_G-1)$
Total	$2nm(n+m-1)$	$n(2mN_G+N_G-m)$

Table 1: Arithmetic operations count to calculate S_F.

In the definition of S_F (3.14), we have that $\forall i, \forall j$, the function to be integrated on the face F is a polynomial of degree $m+n-2$ (assuming that $n \geq m$). Therefore, to obtain an exact integration of this function using a Gauss-Legendre quadrature formula it suffices that :

$$m+n-2 \leq 2N_G-1 \Rightarrow \quad N_G \geq \frac{n+m-1}{2} \Rightarrow \quad N_G = \left[\frac{n+m-1}{2}\right],$$

where $[.]$ indicate the floor function of a real number. We denote by $\mathcal{C}$ the difference of total operations number between the projection and quadrature approaches. Three cases must be considered:

- if $n+m-1$ is an even number, then $N_G = \dfrac{n+m-1}{2}$ and we have :

$$\begin{aligned}\mathcal{C} &= 2nm(n+m-1) - n(2mN_G+N_G-m) \\ &= mn^2 + nm^2 - \frac{nm}{2} - \frac{n^2}{2} + \frac{n}{2} \\ &= n^2(m-\frac{1}{2}) + mn(m-\frac{1}{2}) + \frac{n}{2} \;>\; 0, \forall m, \forall n.\end{aligned}$$

- if $n+m-1$ is an odd number, then $N_G = \dfrac{n+m}{2}$ and we have :

$$\begin{aligned}\mathcal{C} &= 2nm(n+m-1) - n(2mN_G + N_G - m) \\ &= 2mn^2 + 2nm^2 - 2mn - nm^2 - mn^2 - \frac{n^2}{2} + \frac{mn}{2} \\ &= n^2(m - \frac{1}{2}) + mn(m - \frac{3}{2}) \ > \ 0, \forall m, \forall n \text{ such that } (n,m) \neq 1.\end{aligned}$$

- if $n = m = 1$, then $\mathcal{C} = 0$ and the two approaches are equivalent.

We summarize in Tab. 2 the total operation counts for the two approaches and for the most frequently observed cases i.e. $n = m$, $n = m+1$ and $n = m+2$.

Case	Projection approach	Quadrature approach
$n = m$	$2m^2(2m-1)$	$2m^3$
$n = m+1$	$4m^2(m+1)$	$2m^2(m+1)$
$n = m+2$	$(m+2)(4m^2+2m)$	$(m+2)(2m^2+2m+1)$

Table 2: Total operations count to calculate S_F.

Remark 1 *In the case of the non-conforming situation of Fig. 3.6 (a), the choice $P^1 = I$ (the identity matrix) and thus $S_F = A_{F'}(P^2)^t$, is conceivable and makes it possible to reduce the costs of the evaluation of S_F. For the projection approach, we only need the product of two matrices and the cost decreases in this case, see Tab. 3. On the other hand, for the quadrature approach, we always need the product of three matrices and the operation count does not change.*

In summary:

- if $n = m$, the projection approach is less expensive than the quadrature one by an amount of m^2 operations.
- if $n \neq m$, the projection approach is more expensive than the quadrature one.

Consequently, we can conclude that the quadrature approach is always the less expensive.

Operation	Projection approach	Quadrature approach
$\times$	nm^2	$nN_G(m+1)$
$+$	$nm(m-1)$	$nm(N_G-1)$
Total	$nm(2m-1)$	$n(2mN_G + N_G - m)$

Table 3: Operations count to calculate S_F in the case of a classical non-conforming mesh.

Fig. 3.7 shows the CPU time (in seconds) for the evaluation of S_F as a function of the number of faces, using the Gauss-Legendre numerical quadrature approach. The CPU time increases very slowly compared to the number of faces. The Gauss-Legendre quadrature was implemented in Fortran 77 and the computation was performed on a PC workstation equipped with an Intel P4 3.6 GHz processor.

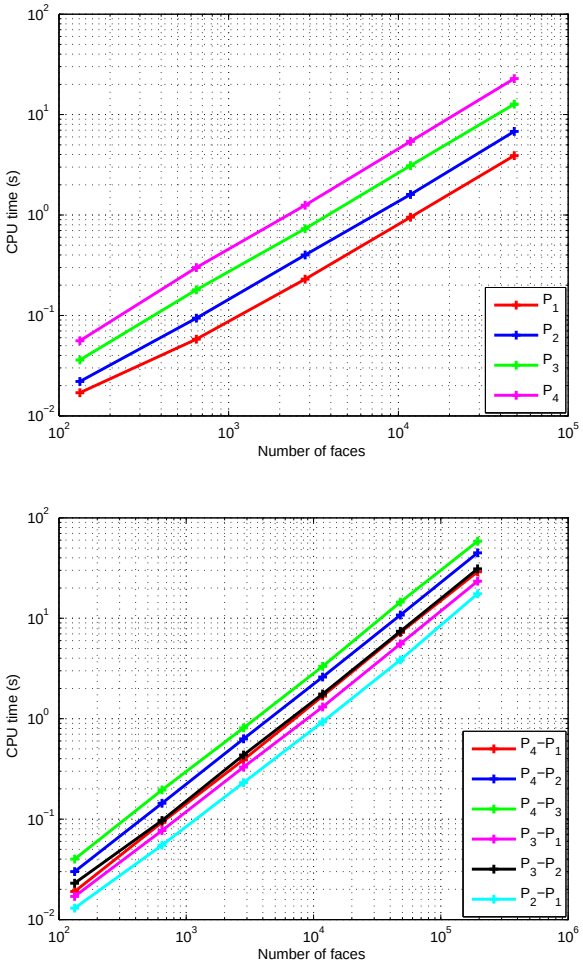


Figure 3.7: The calculation cost of S_F using Gauss-Legendre quadrature approach. For $n = m$ (top) and for $n \neq m$ (bottom) (n being the degree in T_1 and m the degree in T_2).

4 Numerical results for h-refinement and p-enrichment

In this section we present numerical experiments to highlight the practical performance of the h-refinement and p-enrichment versions of the discontinuous Galerkin method using conforming and non-conforming meshes with arbitrary level of hanging nodes. The h-refinement involves a modification of the element size (for a fixed p) while the p-enrichment yields a modification of the polynomial degree (for a fixed h). Our main goal is to verify the energy conservation property of the $\mathbb{P}_{p_i}$-DGTD schemes and to compare them on the basis of the observed numerical dispersion. Note that throughout this section, we suppose that the polynomial degree $p_i = p$ is constant in all elements T_i of the meshes and we use only metallic boundaries conditions. For each case, we shall precise the used numerical CFL conditions.

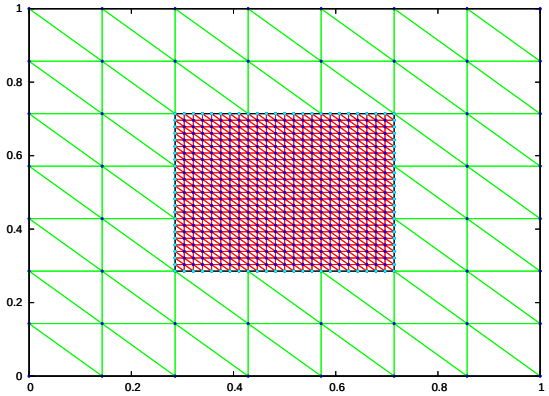


Figure 4.8: 7-irregular triangular mesh

We compute the $(1,1)$ mode which is a standing wave of frequency $f = 0.212$ GHz and wavelength $\lambda = \dfrac{c}{f}$ ($c^2\epsilon\mu = 1$) in a unitary square metallic cavity. The exact solution is known thus allowing us to appreciate the numerical results at any point and time in the cavity. We initialize the electromagnetic field by the analytical solution at $t = 0$, given by the following expression: $H_x = H_y = 0$ and $E_z = \sin(\pi x)\sin(\pi y)$. The exact solution is given by:

$$\begin{cases} H_x &= -\dfrac{1}{2f}\sin(\pi x)\cos(\pi y)\sin(2\pi f)t, \\ H_y &= \dfrac{1}{2f}\cos(\pi x)\sin(\pi y)\sin(2\pi f)t, \\ E_z &= \sin(\pi x)\sin(\pi y)\cos(2\pi f)t. \end{cases}$$

We propose first to verify the stability of the $\mathbb{P}_p$-DGTD schemes, the objective being to assess the stability of the $\mathbb{P}_p$-DGTD schemes in presence of a high level of h-refinement. In order to do so, we first construct a regular triangular mesh consisting of 200 triangles, 320 faces and 121 nodes which gives 10 points per wavelength. Then, an irregular mesh is obtained by locally refining three times the regular mesh. The resulting irregular mesh consists of 1232 triangles (1152 of them in the fine grid), 1904 faces (1680 of them in the fine grid, 100 in the coarse grid and 96 non-conforming faces) and 673 nodes (84 of which are hanging nodes i.e. we allow 7-level hanging nodes on a face, see Fig. 4.8). We use for each p the maximal allowable time step for a stable simulation. All simulations are carried out for 43 periods. We summarize in Tab. 4 the corresponding (numerical)

CFL values. We plot on Fig. 4.9 and 4.10 the time evolution of the energy for the $\mathbb{P}_p$-DGTD schemes, $p = 0, \ldots, 4$ on the regular and 7-irregular meshes, respectively. Clearly, all the schemes are stable even in the presence of a high level of non-conforming refinement. Moreover, one can note that the $\mathbb{P}_p$-DGTD energy for $p \geq 2$ compares very well with the exact one.

p	0	1	2	3	4
CFL	0.7	0.3	0.15	0.1	0.09

Table 4: Numerical CFL values for the $\mathbb{P}_p$-DGTD schemes.

Since the exact solution is known, we can evaluate the accuracy of the $\mathbb{P}_p$-DGTD schemes from the point of view of the observed numerical dispersion. This time, we make use of different meshes depending on the value of the interpolation order p. First, a coarse and a fine regular meshes are constructed whose characteristics are summarized in Tab. 5. Then, 7-irregular meshes are obtained by locally refining three times a rectangular region in the center of the regular meshes. The characteristics of the resulting 7-irregular meshes are given in Tab. 6. As previously the simulations are carried out for 43 periods. We plot on Fig. 4.11 and 4.12 the time evolution of H_x at a fixed point in the coarse mesh for each $\mathbb{P}_p$-DGTD scheme. One can see that for both regular and irregular meshes, the $\mathbb{P}_p$-DGTD ($p \geq 2$) solutions compares very well with the exact one. On the other hand, the $\mathbb{P}_0$ and $\mathbb{P}_1$-DGTD solutions exhibit a large dispersive error for both the regular and irregular meshes. Despite that, for irregular meshes the solutions seem more accurate. The overall L^2 error (L^2 norm of the difference between the approximate and exact solutions) on the electromagnetic field (H_x, H_y, E_z) of the $\mathbb{P}_p$-DGTD approximate solutions are plotted on Fig. 4.13. The errors are increasing in time because of the dispersion and the level of dispersion is notably lower for the $\mathbb{P}_p$-DGTD, $p \geq 2$. Finally, Fig. 4.14 shows the x-wise distributions for $y = 7.5$ of the H_x component for the exact and approximate ($p \leq 2$) solutions.

	# triangles	# faces	# nodes
$p = 0, 1$	288	456	169
$p = 2, 3, 4$	72	120	49

Table 5: Characteristics of the regular meshes for each p.

	# coarse triangles	# fine triangles	# conforming faces	# non-conforming faces	# nodes	# hanging nodes
$p = 0, 1$	256	2048	3408	128	1121	112
$p = 2, 3, 4$	64	512	840	64	273	56

Table 6: Characteristics of the 7-irregular meshes for each p.

In summary, we can conclude that:

- for a fixed mesh size h, increasing the polynomial degree p can reduce the dispersion error for $p \geq 2$.
- for a fixed polynomial degree p, when locally refining the mesh and allowing hanging nodes, the accuracy of the solution is preserved and the dispersion error is reduced.
- for $p = 0, 1$, the solution exhibits a large dispersion even if the mesh is locally refined.

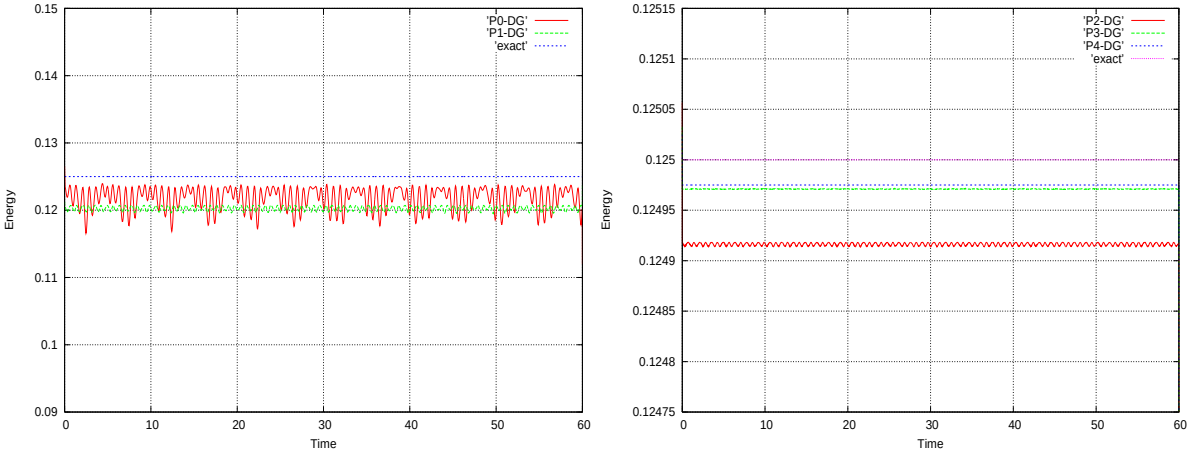


Figure 4.9: Time evolution of the energy: $\mathbb{P}_p$-DGTD schemes, $p = 0, \ldots, 4$, regular meshes.

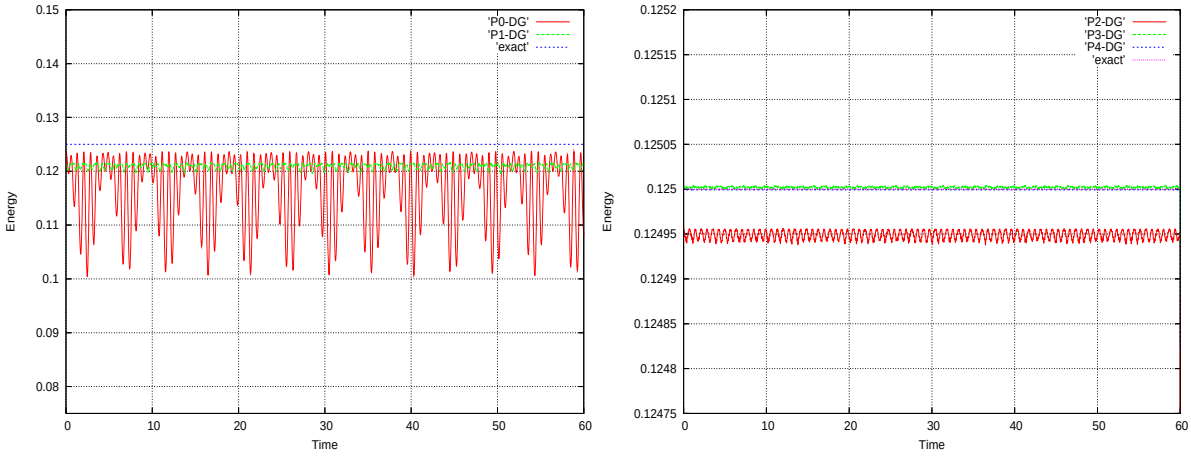


Figure 4.10: Time evolution of the energy: $\mathbb{P}_p$-DGTD schemes, $p = 0, \ldots, 4$, irregular meshes.

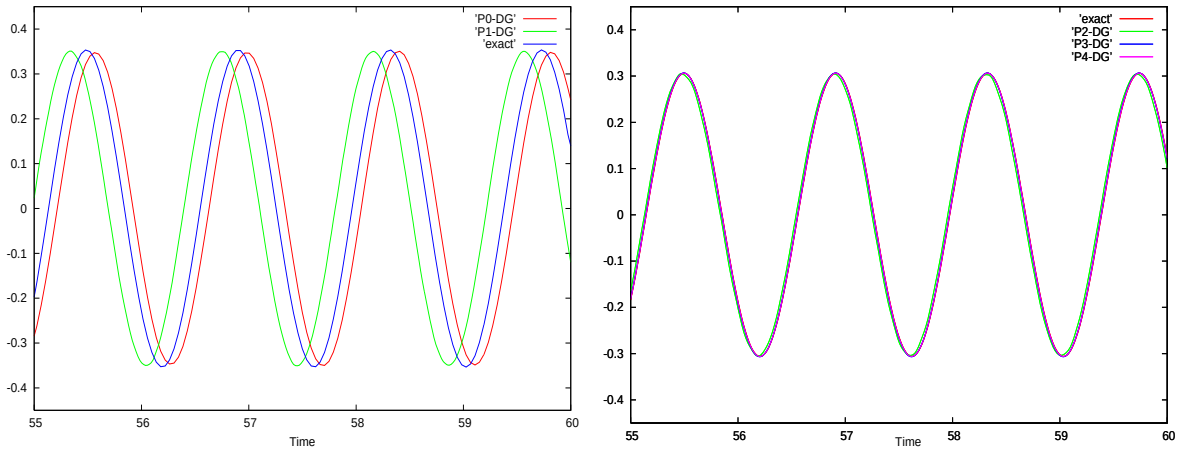


Figure 4.11: Time evolution of H_x: $\mathbb{P}_p$-DGTD schemes and exact solution, regular meshes. Zoom on the last 4 periods.

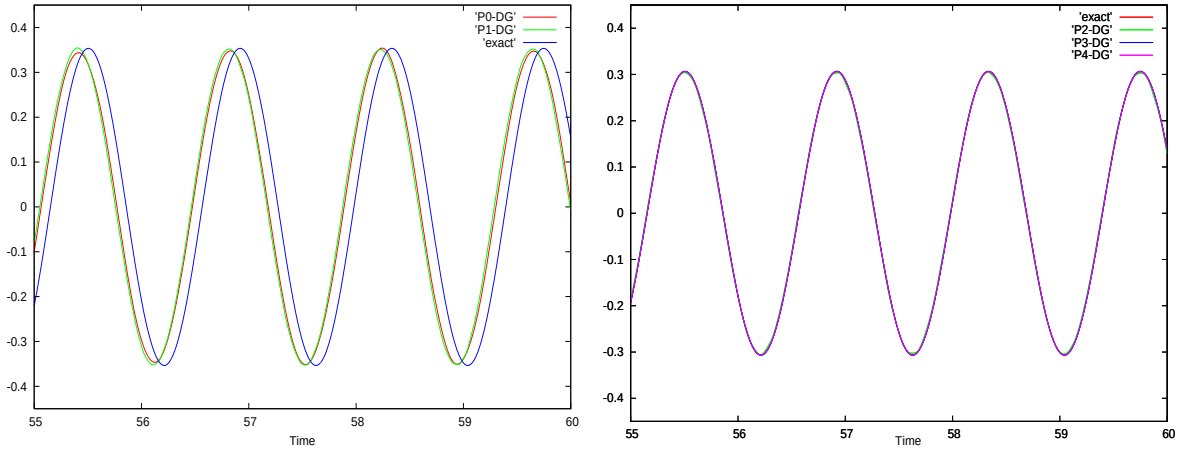


Figure 4.12: Time evolution of H_x: $\mathbb{P}_p$-DGTD schemes and exact solution, irregular meshes. Zoom on the last 4 periods.

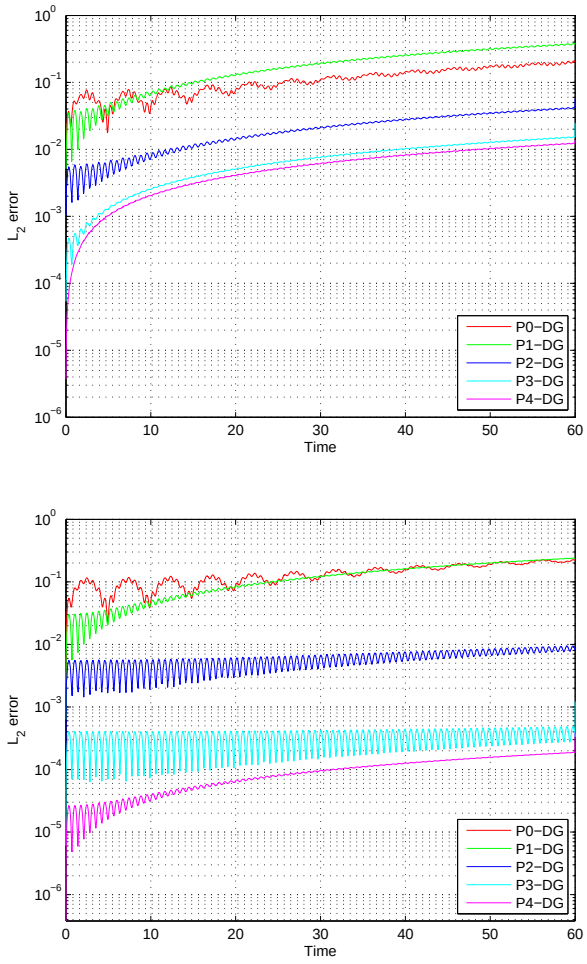


Figure 4.13: Time evolution of the L^2 error: $\mathbb{P}_p$-DGTD schemes.
Logarithmic scale on the y-axis.
Regular (top) and irregular (bottom) meshes.

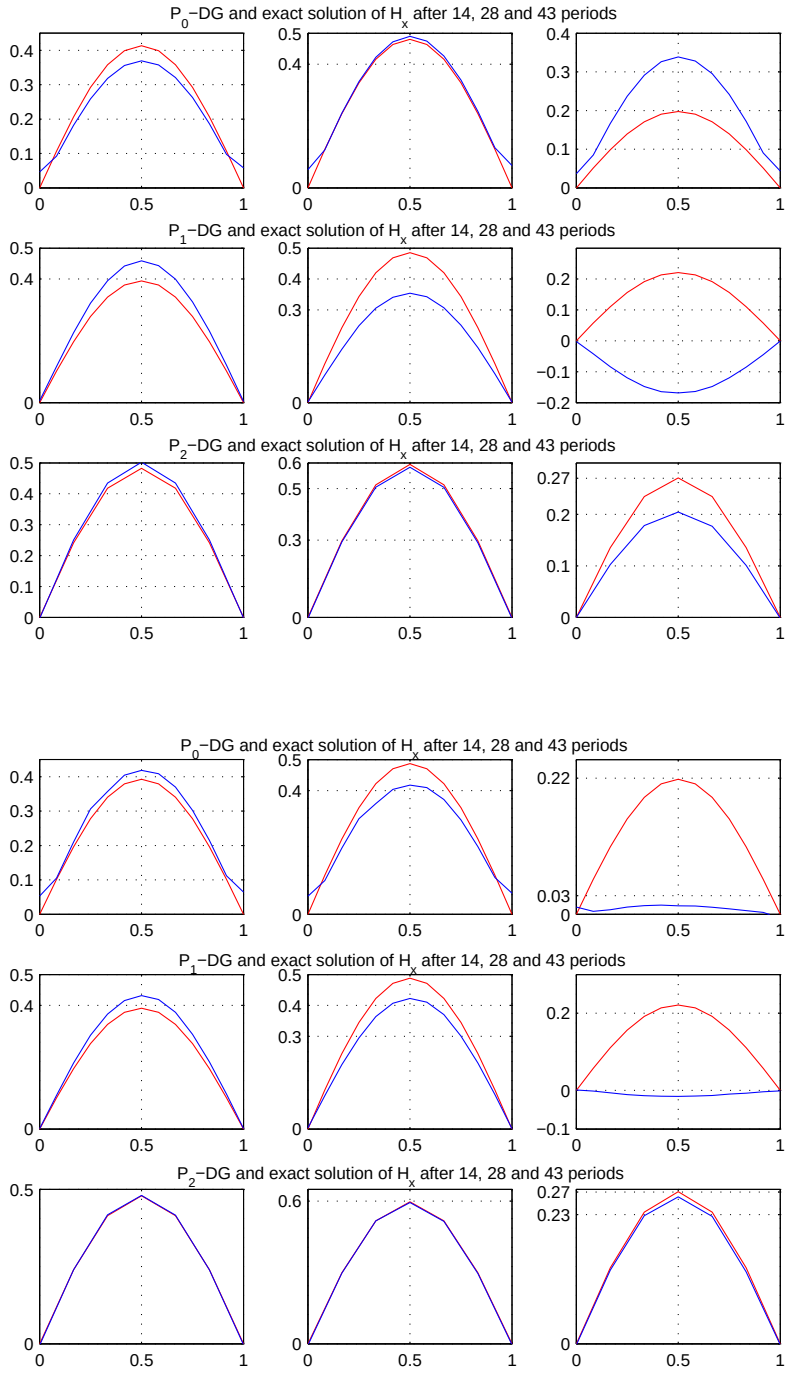


Figure 4.14: 1D distributions of H_x: exact (in blue) and $\mathbb{P}_p$-DGTD solutions ($p \leq 2$), regular (top) and irregular (bottom) meshes.

In the next section, we propose an approach which combines h-refinement with p-enrichment to further reduce the dispersion error of these $\mathbb{P}_p$-DGTD schemes, among other objectives.

5 A hp-like discontinuous Galerkin method

In this section we propose a hp-like DGTD method in which we combine h-refinement and p-enrichment. We have seen in section 4 that the dispersion error is not reduced when h-refinement is applied for $\mathbb{P}_p$-DGTD ($p = 0, 1$) schemes. Moreover, the p-enrichment requires a large number of degrees of freedom, and additional CPU time and memory costs, which increase with p. The main objectives of the hp-like DGTD method studied here are to eliminate or to strongly reduce the dispersion error, and to decrease the computational cost.

In order to improve the representation of the fields on the refined mesh, the hp-like DGTD method consists in using high order polynomials in the coarse mesh and low order polynomials in the fine one (e.g. the refined elements). We name the resulting scheme the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method, where p_c and p_f are the polynomial degrees in the coarse and fine elements respectively. If $p_c = p_f = p$, we recover the classical $\mathbb{P}_p$-DGTD schemes studied in section 4 (also in [18]). Fig. 5.15 shows two different representations of the fields, the first one using $\mathbb{P}_2$ and $\mathbb{P}_1$ approximations and the second one using $\mathbb{P}_3$ and $\mathbb{P}_0$ approximations. The stability of such a $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method is ensured under some CFL conditions. In Tab. 7, we summarize the CFL conditions numerically observed for some polynomial degrees. These CFL conditions are to be compared with those obtained for the $\mathbb{P}_p$-DGTD scheme (see Tab. 4). One can see that for $p_c = p_f + 1 = p + 1$, the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method has the same stability limit as the $\mathbb{P}_p$-DGTD scheme, as long as the mesh is actually refined. This is not a surprise, since the $\mathbb{P}_{p_c}$-DGTD scheme, which has a reduced stability domain, is only used on elements of the coarse mesh (which are at least twice larger than elements of the refined mesh).

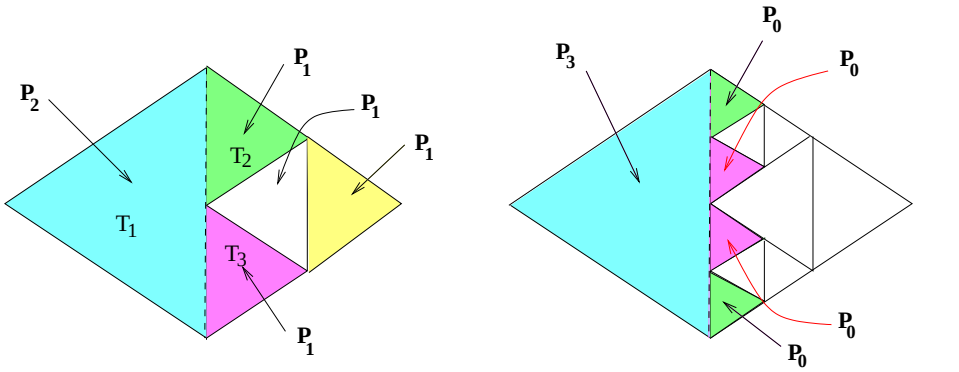


Figure 5.15: Fields representation examples.

In order to validate the method we again consider the square resonant cavity and we apply various $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes to compute the time evolution of the $(1, 1)$-mode using the 7-irregular (non-conforming) triangular mesh of Fig. 4.8. The number of points per wavelength λ/h is equal to 6 in the coarse mesh (without refinement) and 48 in the refined zone, see Tab. 8. The degrees

Scheme	$\mathbb{P}_1$:$\mathbb{P}_0$	$\mathbb{P}_2$:$\mathbb{P}_0$	$\mathbb{P}_3$:$\mathbb{P}_0$	$\mathbb{P}_4$:$\mathbb{P}_0$	$\mathbb{P}_2$:$\mathbb{P}_1$
CFL	0.7	0.4	0.3	0.18	0.3
Scheme	$\mathbb{P}_3$:$\mathbb{P}_1$	$\mathbb{P}_4$:$\mathbb{P}_1$	$\mathbb{P}_3$:$\mathbb{P}_2$	$\mathbb{P}_4$:$\mathbb{P}_2$	$\mathbb{P}_4$:$\mathbb{P}_3$
CFL	0.25	0.18	0.15	0.15	0.1

Table 7: Numerical CFL conditions for $\mathbb{P}_{pc}$:$\mathbb{P}_{pf}$-DGTD schemes.

of freedom are initialized by the projection of the exact solution on the local basis functions. We show on Fig. 5.16 the time evolution of the energy for some polynomial degrees p_c and p_f after 43 periods. One can clearly see that the energy is conserved. On Fig. 5.17 and 5.18 we compare the time evolution of the approximate and exact values of the H_x component at a given point in the coarse grid (the five last periods of 43 are shown) for the $\mathbb{P}_{p_c}$:$\mathbb{P}_0$-DGTD and $\mathbb{P}_{p_c}$:$\mathbb{P}_1$-DGTD ($p_c \geq 2$) schemes. We note that these schemes do not introduce too much dispersion in the coarse grid where the CFL is far from optimal. Although the level of refinement is high, the coarse mesh is characterized by a few points per wavelength and the simulation is quite long. As far as $\mathbb{P}_1$:$\mathbb{P}_0$-DGTD scheme is concerned, we can observe that it is dispersive (see Fig. 5.19) which is not a surprise because as we mentioned in section 4 the $\mathbb{P}_1$-DGTD and $\mathbb{P}_0$-DGTD schemes have an important dispersion error, then it is obvious that the $\mathbb{P}_1$:$\mathbb{P}_0$-DGTD scheme will be dispersive. We show on Fig. 5.20 x-wise distributions for $y = 7.5$ of the H_x component for the exact and approximate solutions. These curves have to be compared with those of Fig. 4.14. One can note that the solutions provided by the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes compare very well with those of the standard $\mathbb{P}_p$-DGTD schemes. The time evolution of the L^2 error on the electromagnetic field (H_x, H_y, E_z) computed with the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes are plotted on Fig. 5.21. We can observe that it is not necessary to increase the polynomial degree in the coarse mesh to more than three. Fig. 5.21 is to be compared with Fig. 4.13 (bottom) which corresponds to the $\mathbb{P}_p$-DGTD schemes (i.e. $p_c = p_f = p$). It appears that the $\mathbb{P}_{p_c}$:$\mathbb{P}_1$-DGTD method ($p_c = 3, 4$) is more accurate (in terms of the L^2 error) than the $\mathbb{P}_2$-DGTD while the latter is comparable with the $\mathbb{P}_{p_c}$:$\mathbb{P}_0$-DGTD schemes ($p_c = 3, 4$). One can see also that the $\mathbb{P}_3$:$\mathbb{P}_2$-DGTD scheme has a similar L^2 error level as the $\mathbb{P}_p$-DGTD ($p \geq 3$) schemes.

# coarse triangles	# fine triangles	# conforming faces	# non-conforming faces	# nodes	# hanging nodes
64	512	840	64	273	56

Table 8: Characteristics of the 7-irregular mesh.

Finally, some conclusions can be drawn from this numerical study:

- the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method can reduce (or eliminate) the dispersion error which derives from the $\mathbb{P}_p$-DGTD schemes, for $p = 0, 1$.
- increasing the polynomial degree higher than three in the coarse mesh does not improve too much the result.

In the next section, we will study the numerical convergence and the computational cost of all the methods studied so far.

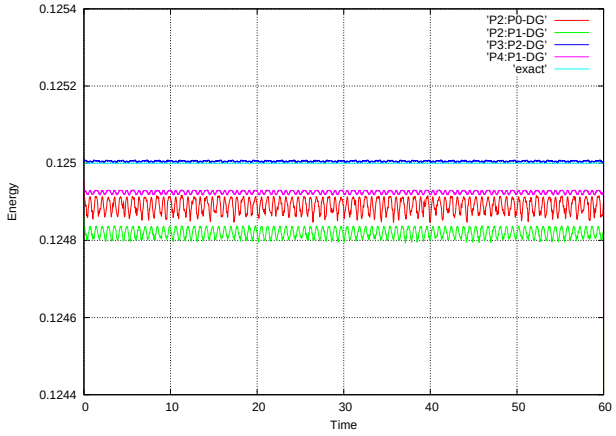
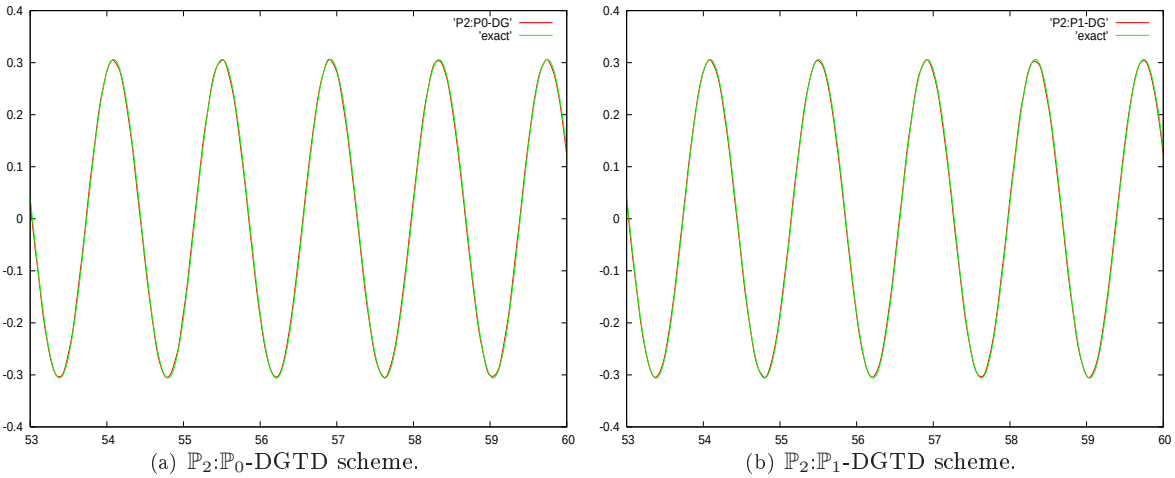
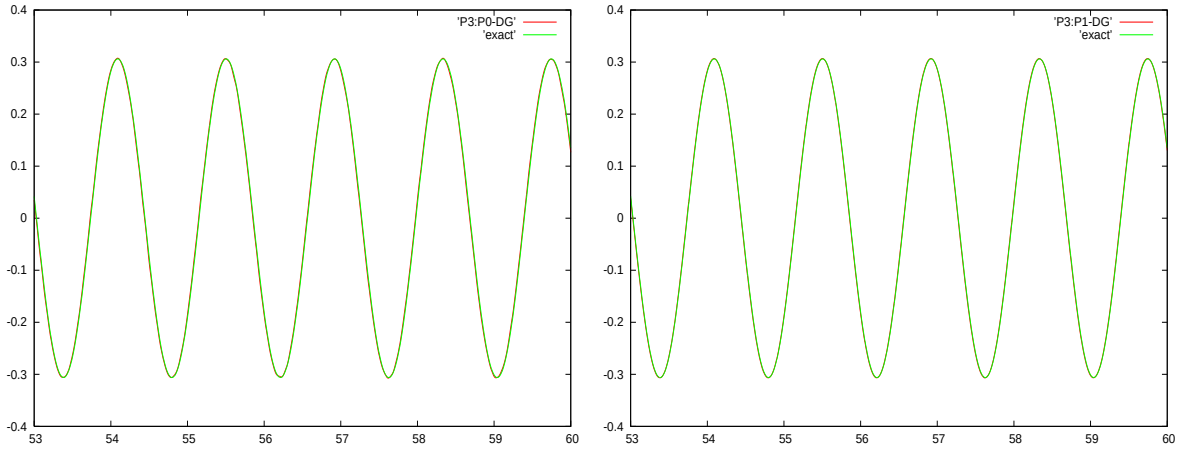


Figure 5.16: Time evolution of the energy: $\mathbb{P}_{pc}$:$\mathbb{P}_{pf}$-DGTD schemes.



(a) $\mathbb{P}_2$:$\mathbb{P}_0$-DGTD scheme. (b) $\mathbb{P}_2$:$\mathbb{P}_1$-DGTD scheme.

Figure 5.17: Time evolution of H_x: zoom on the last five periods, 7-irregular mesh.



(a) $\mathbb{P}_3$:$\mathbb{P}_0$-DGTD scheme. (b) $\mathbb{P}_3$:$\mathbb{P}_1$-DGTD scheme.

Figure 5.18: Time evolution of H_x: zoom on the last five periods, 7-irregular mesh.

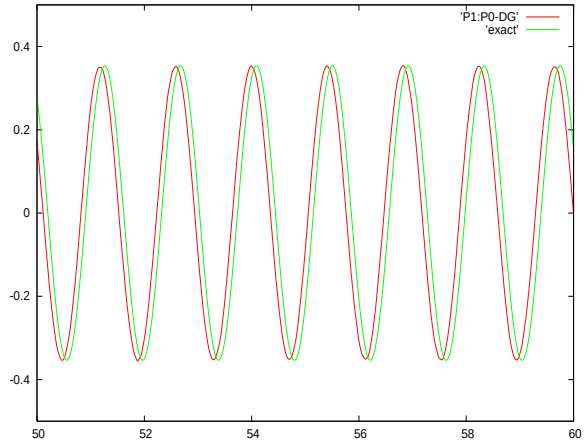


Figure 5.19: Time evolution of H_x: zoom on the last seven periods, 7-irregular mesh. $\mathbb{P}_1$:$\mathbb{P}_0$-DGTD scheme.

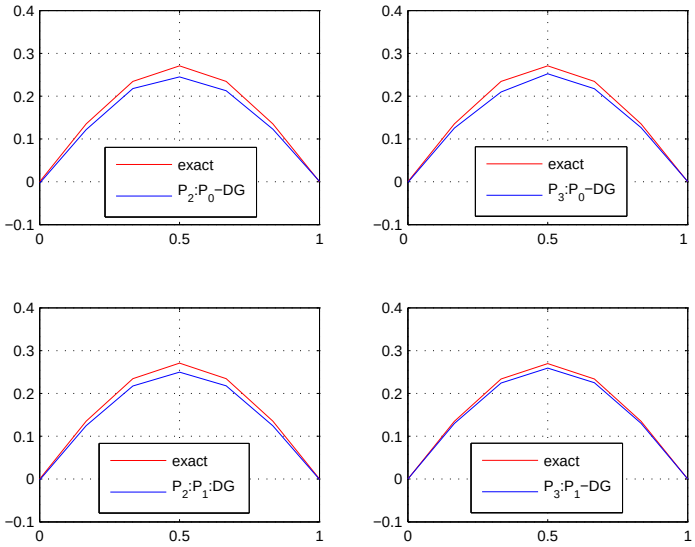


Figure 5.20: 1D distributions of H_x: exact and $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD solutions after 43 periods. 7-irregular mesh.

6 Numerical convergence

In this section, we study numerically the convergence of the schemes suggested in sections 4 and 5: p-enrichment for a fixed h, h-refinement for a fixed p and combined hp-refinement. In addition to the convergence of the various schemes, we pay attention to the CPU time.

A theoretical convergence analysis of the $\mathbb{P}_{p_i}$-DGTD method is conducted in [19], in the case of conforming simplicial meshes and $p_i = p$ everywhere. It is shown that the convergence order in space and time is:

$$\mathcal{O}(Th^{\min(s,p)}) + \mathcal{O}(\Delta t^2) \tag{6.17}$$

where $t \in [0, T]$ and s is such that the solution is sufficiently regular. One can note that the asymptotic convergence order is bounded by 2 independently of the interpolation degree p and the formula (6.17) seems to be sub-optimal. This comes from the fact that the leap-frog time-scheme is second-order accurate. According to the authors, the formula (6.17) applies equally to non-conformal refinement (h-refinement) including hp-adaptive method. Our attention is turned to the validity of this result especially in the cases of the h-refinement, p-enrichment and combined hp-refinement DGTD methods proposed in this work with non-conforming meshes.

In the following, by *global refinement* we mean a mesh refined in a conforming way i.e. the various meshes are obtained by increasing several times the number of points per wavelength of a basic conforming mesh. By *local refinement* we mean a mesh refined in a non-conforming way i.e. we fix a basic mesh which is locally refined three times in the center zone $[0.35, 0.65]^2$ and we obtain the various meshes by increasing several times the number of points per wavelength inside and outside the refined zone. We denote by DOF the number of degrees of freedom in the mesh and the CPU time is measured in seconds. *Global convergence order* means the coefficient a (slope) of the regression problem $y = ax + b$, such that x and y are respectively the Neperian logarithms

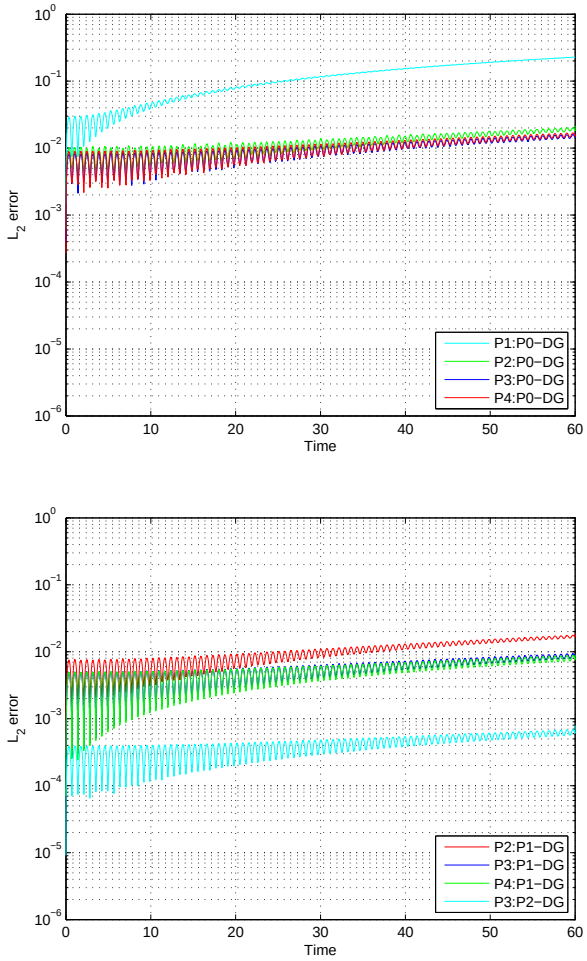


Figure 5.21: Time evolution of the L^2 error: $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes.
Logarithmic scale on the y-axis.
7-irregular mesh.

of the vectors which contain the values of $\sqrt{\text{DOF}}$ and the L^2 error. *Efficiency order* refers to the coefficient a in $y = ax+b$ such that x and y are respectively the Neperian logarithms of the vectors which contain the values of $\sqrt{\text{DOF}}$ and the CPU time. By *order* (or local convergence order) we mean the coefficient a evaluated using two points of abscissa $\sqrt{\text{DOF}}$ and of ordinate the L^2 error. Finally, the numerical convergence studies presented here still concern the test case already adopted in sections 4 and 5.

6.1 p-enrichment DGTD method

Let us begin by considering the case where h is fixed and the polynomial degree p is varied. To illustrate the convergence properties of the corresponding DGTD scheme, we summarize in Tab. 9 and 10 the L^2 errors, the numerically observed convergence order, the CPU time and the number of time steps, for $p = 0, \ldots, 4$, using local and global refinements, after two periods. The results of Tab. 9 are plotted on Fig. 6.22 and 6.23. Fig. 6.22 shows the convergence curves for three different conforming and non-conforming meshes. They clearly indicate different convergence orders. We have calculated the global convergence orders observed by using logarithmic linear regression (see the bottom of Tab. 9). These orders of convergence can be interpreted as follows: first of all, in the case of globally refined meshes, the global convergence order of the p-enrichment DGTD method is limited to 4.8; for locally refined meshes, the global convergence order of the p-enrichment DGTD method reaches 7.8. In addition, it seems to exist an exponential convergence phenomenon if we limit the polynomial degree to 2 (i.e. $p \leq 2$).

Local refinement					Global refinement				
p	$\sqrt{\text{DOF}}$	L^2 error	order	CPU (s)	p	$\sqrt{\text{DOF}}$	L^2 error	order	CPU (s)
Mesh1 (# (Face)$_{\text{nc}} = 72$)					Mesh1				
0	19	2.84E−01	–	< 1	0	13	6.77E−02	–	< 1
1	32	1.10E−01	1.7	< 1	1	22	3.97E−02	0.9	< 1
2	45	7.11E−03	7.9	4	2	31	1.17E−03	10.2	< 1
3	58	3.82E−04	11.4	10	3	40	4.36E−04	3.9	1
4	71	4.66E−05	10.4	19	4	49	3.60E−04	0.9	2
Mesh2 (# (Face)$_{\text{nc}} = 120$)					Mesh2				
0	31	8.10E−02	–	< 1	0	26	3.33E−02	–	< 1
1	53	3.62E−02	1.5	4	1	44	1.09E−02	2.03	1
2	75	2.13E−03	8.2	18	2	62	2.74E−04	10.6	5
3	97	5.08E−05	14.6	53	3	80	1.37E−04	2.7	10
4	119	6.43E−06	10.2	104	4	98	1.18E−04	0.8	19
Mesh3 (# (Face)$_{\text{nc}} = 264$)					Mesh3				
0	66	5.82E−02	–	8	0	52	2.02E−02	–	1
1	114	9.94E−03	3.2	45	1	88	2.78E−03	3.6	10
2	162	1.37E−04	12.4	176	2	124	9.40E−05	9.8	40
3	210	3.91E−05	4.9	489	3	160	6.21E−05	1.6	81
4	258	1.26E−06	16.9	955	4	196	5.74E−05	0.4	155
Global convergence order					Global convergence order				
	Mesh1	Mesh2	Mesh3			Mesh1	Mesh2	Mesh3	
	6.5	7.1	7.8			4.5	4.7	4.8	

Table 9: Convergence study for the p-enrichment DGTD method using local and global refinements. L^2 errors and CPU times are measured after two periods.
(Face)$_{\text{nc}}$ represents the number of non-conforming interfaces.

Local refinement				Global refinement			
p	Mesh1	Mesh2	Mesh3	p	Mesh1	Mesh2	Mesh3
0	229	458	916	0	65	130	260
1	534	1068	2136	1	150	300	600
2	1068	2136	4272	2	300	600	1200
3	1600	3200	6400	3	450	900	1800
4	1778	3556	7112	4	500	1000	2000

Table 10: Number of time steps for a two-period duration. p-enrichment DGTD method using local and global refinements.

Fig. 6.23 shows the CPU time as a function of the number of degrees of freedom for both a local and a global refinement of the meshes. We obtain different efficiency curves. We have calculate the global efficiency orders, these are given in Tab. 11.

Local refinement			Global refinement		
Mesh1	Mesh2	Mesh3	Mesh1	Mesh2	Mesh3
3.74	3.81	3.63	3.68	3.73	3.71

Table 11: Efficiency orders for the p-enrichment method using local and global refinements.

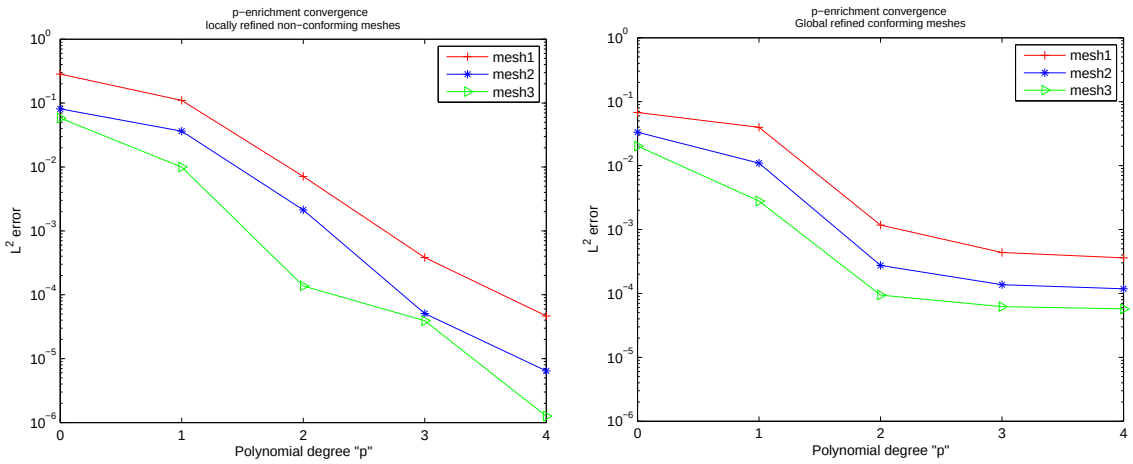


Figure 6.22: L^2-error as a function of the polynomial degree p for different meshes. Local refinement (left) and global refinement (right).

A trivial consequence from Tab. 9 is that to obtain a fixed error, the CPU time with local refinement is more expensive than using a global one, despite that the local refinement based DGTD method has a higher convergence rate. One can also conclude that the only additional CPU time resulting from the non-conformity is the cost of using Gaussian quadrature to evaluate the non-conforming interface matrix (2.10). Indeed, the Gauss-Legendre quadrature is only used to evaluate the non-conforming interface matrix, while the conforming interface matrices are calculated one time for all. To make this point clear, we consider the case of Mesh3 for $p = 3$ in Tab. 9 and 10. In this case the p-enrichment method requires 264 non-conforming interfaces, 6400 time steps and 489 seconds

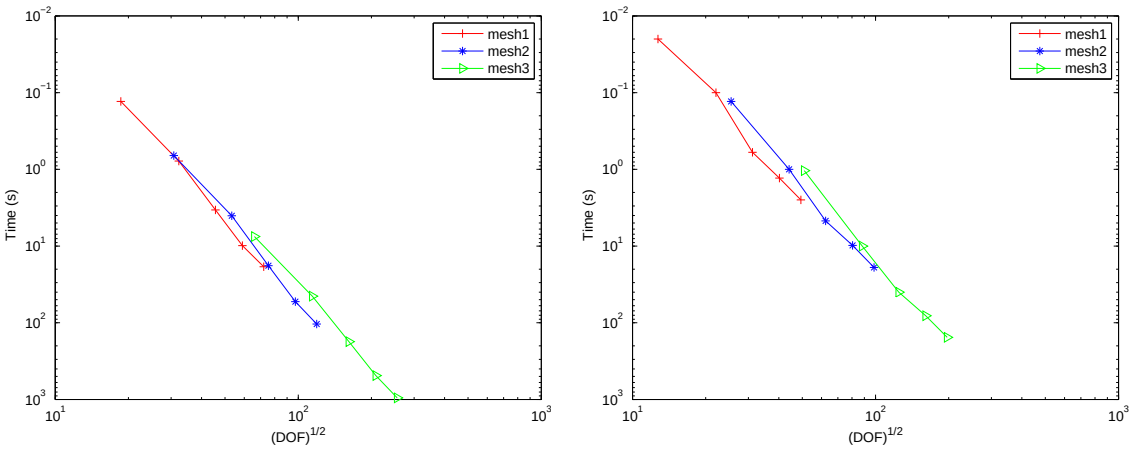


Figure 6.23: CPU time as a function of $\sqrt{\text{DOF}}$ for different meshes.
Local refinement (left) and global refinement (right).

for local refinement, and 1800 time steps and 81 seconds for global refinement. Then, the CPU cost of a time step is 0.08 s for local refinement and 0.05 s for global refinement. The additional CPU cost is to be compared with the CPU cost of the 264 non-conforming interface matrices which is plotted on Fig. 3.7 (top) and is found to be 0.04 s (not far from the 0.03 s observed here).

6.2 h-refinement DGTD method

Let us consider the case where the interpolation order p is uniform and the mesh size h is varied. To illustrate the convergence properties of the corresponding DGTD scheme, we summarize in Tab. 12 and 13 the L^2 errors, the numerically observed convergence order, the CPU time and the number of time steps, for $p = 0, \ldots, 3$ and using local and global refinements, after two periods. The results of these tables are plotted on Fig. 6.24 to 6.26. Fig. 6.24 shows the convergence curves that clearly indicate different convergence orders. We have calculated the global convergence orders observed by using logarithmic linear regression (see the bottom of Tab. 12 and 13). We note that the global convergence order is essentially bounded by 2 (excepted for $p = 2$ and non-conforming meshes for which we obtain 2.2) which confirms the theoretical convergence result (6.17). One can observe, that for $p = 0, 1$ we obtain a convergence order of rate $\mathcal{O}(h^{p+1})$ and from this point of view, it seems that the formula (6.17) is sub-optimal.

Fig. 6.25 shows the L^2 error as a function of the CPU time for local and global refinements. One can see that for any mesh case and in the two-dimensional case, the gains of computational time are notable if we increase the accuracy of the schemes in space. In the three-dimensional case, these gains could be less clear because the number of degrees of freedom increases faster with the polynomial degree p.

Finally, we show on Fig. 6.26 the CPU time as a function of the number of degrees of freedom for both local (left) and global (right) refinement. We have calculated the global efficiency orders and these are given in Tab. 14.

A detailed analysis of Tab. 12 and 13 motivates the following remarks:

Local refinement				
$\sqrt{\text{DOF}}$	L^2 error	order	CPU (s)	# time steps
		$\mathbb{P}_0$-DG		
48	7.22E−02	–	3	700
96	3.61E−02	0.99	29	×2
192	1.82E−02	0.99	249	×4
384	9.30E−03	0.97	1957	×8
		$\mathbb{P}_1$-DG		
41.5	5.29E−02	–	2	800
83	1.52E−02	1.80	15	×2
166	3.98E−03	1.93	154	×4
332	1.05E−03	1.92	1266	×8
		$\mathbb{P}_2$-DG		
29.4	1.14E−02	–	1	800
58.8	1.85E−03	2.62	8	×2
117.6	1.98E−04	3.22	68	×4
235.2	4.69E−05	2.08	580	×8
470.4	3.80E−05	0.30	4577	×16
		$\mathbb{P}_3$-DG		
24	5.65E−03	–	1	800
48	3.83E−04	3.88	9	×2
96	7.08E−05	2.43	52	×4
192	4.61E−05	0.62	489	×8
384	3.36E−05	0.46	3707	×16
Global convergence order				
$\mathbb{P}_0$-DG	$\mathbb{P}_1$-DG	$\mathbb{P}_2$-DG	$\mathbb{P}_3$-DG	
0.99	1.89	2.17	1.78	

Table 12: Convergence study for the h-refinement DGTD method using local refinement. L^2 errors and CPU times are measured after two periods.

Global refinement				
$\sqrt{\text{DOF}}$	L^2 error	order	CPU (s)	# time steps
		$\mathbb{P}_0$-DG		
13	6.41E−02	−	< 1	66
26	3.59E−02	0.84	< 1	×2
52	1.68E−02	1.09	2	×4
104	1.04E−02	0.69	17	×8
208	5.25E−03	0.99	133	×16
		$\mathbb{P}_1$-DG		
22	3.97E−02	−	< 1	150
44	1.09E−02	1.87	1	×2
88	2.78E−03	1.97	10	×4
176	7.38E−04	1.91	91	×8
352	2.09E−04	1.82	534	×16
		$\mathbb{P}_2$-DG		
16	1.18E−02	−	< 1	300
31	1.17E−03	3.94	< 1	×2
62	2.73E−04	2.10	5	×4
124	9.40E−05	1.54	40	×8
248	5.13E−05	0.87	324	×16
		$\mathbb{P}_3$-DG		
21	1.35E−03	−	< 1	450
40	4.36E−04	1.92	1	×2
80	1.36E−04	1.67	10	×4
160	6.20E−05	1.14	81	×8
320	3.53E−05	0.81	658	×16
Global convergence order				
$\mathbb{P}_0$-DG	$\mathbb{P}_1$-DG	$\mathbb{P}_2$-DG	$\mathbb{P}_3$-DG	
0.90	1.90	1.98	1.37	

Table 13: Convergence study for the h-refinement DGTD method using global refinement. L^2 errors and CPU times are measured after two periods.

Local refinement			
$\mathbb{P}_0$-DG	$\mathbb{P}_1$-DG	$\mathbb{P}_2$-DG	$\mathbb{P}_3$-DG
3.11	3.13	3.05	2.95

Global refinement			
$\mathbb{P}_0$-DG	$\mathbb{P}_1$-DG	$\mathbb{P}_2$-DG	$\mathbb{P}_3$-DG
3.18	3.13	3.08	2.99

Table 14: Efficiency orders for the h-refinement DGTD method using local and global refinements.

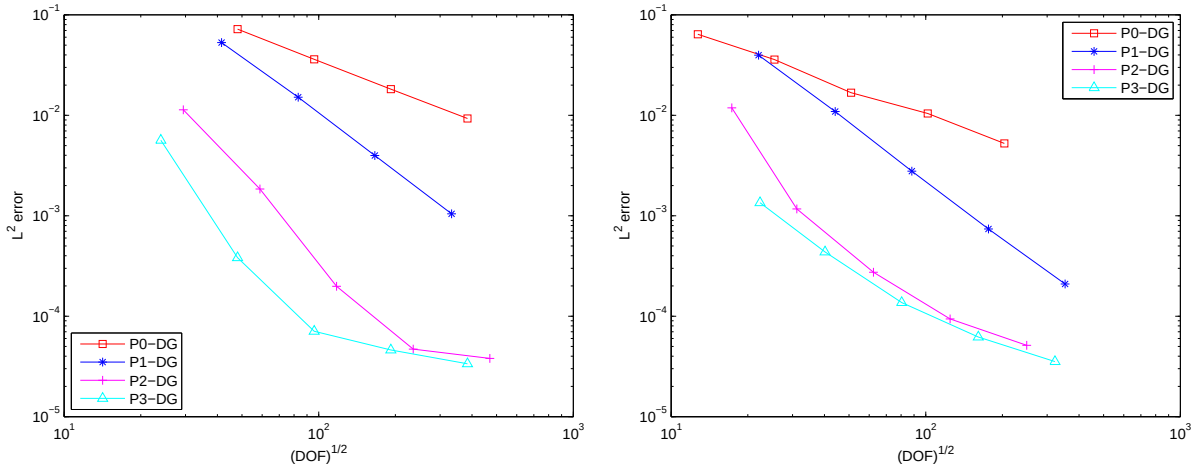


Figure 6.24: L^2 error as a function of $\sqrt{\text{DOF}}$ for the different $\mathbb{P}_p$-DGTD schemes. Local refinement (left) and global refinement (right).

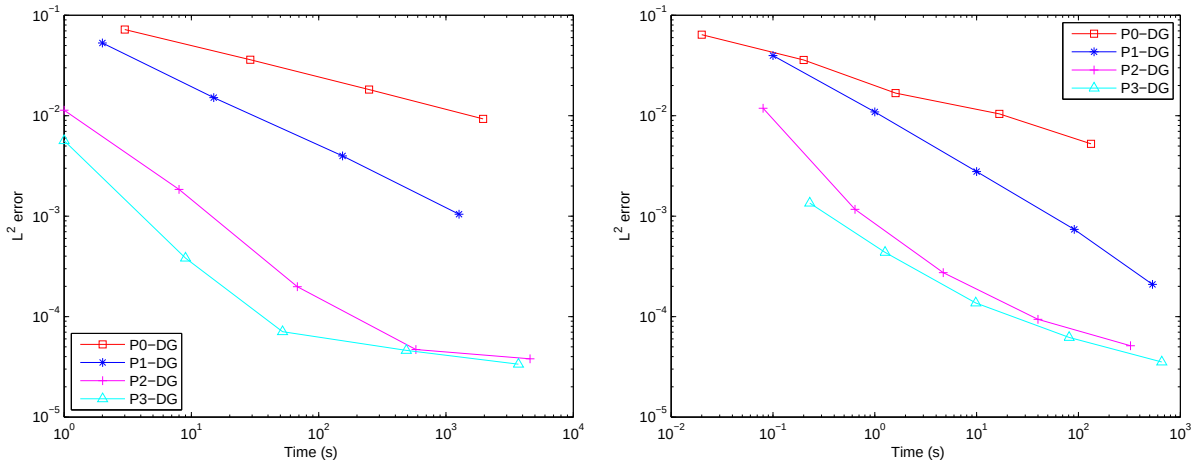


Figure 6.25: L^2 error as a function of the CPU time for the different $\mathbb{P}_p$-DGTD schemes. Local refinement (left) and global refinement (right).

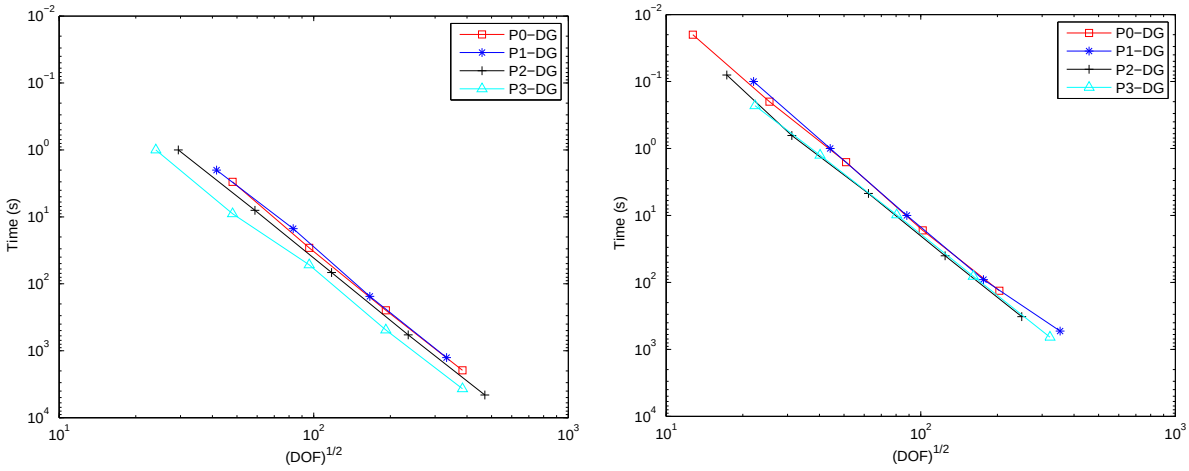


Figure 6.26: CPU time as a function of $\sqrt{\text{DOF}}$ for the different $\mathbb{P}_p$-DGTD schemes. Local refinement (left) and global refinement (right).

- in the case of locally refined meshes and for $p \geq 2$, it seems not reasonable to increase the number of degrees of freedom because, on the one hand, the convergence rate is not improved and, on the other hand, the CPU time increases dramatically, e.g. for $p = 2$, when $\sqrt{\text{DOF}} = 235.2$ (respectively, 470.4) the L^2 error is equal to 4.7E−05 (respectively, 3.8E−05) with a CPU time of 10 mn (respectively, 1 h15 mn).
- in the case of globally refined meshes, it is clear that the convergence rate is not influenced by the increase of the number of DOF and for $p \leq 1$ the h-refinement method is more accurate and less expensive than the local refinement case. For $p \geq 2$ this gain in accuracy and CPU time is less clear compared to the local refinement case.

6.3 *hp*-refinement DGTD method

We now consider the case where both p and h are varied. To demonstrate the performance of this method, we summarize in Tab. 15 the L^2 error, the numerically observed convergence order, the CPU time, the number of time steps and the percentage of degrees of freedom used in the coarse region, after two periods and using locally refined non-conforming meshes. The results of this table are plotted on Fig. 6.27 and 6.28. We show on Fig. 6.27 the convergence curves for different schemes. They indicate different convergence orders and one can note that for $p_f = 0, 1$ (polynomial degrees in the refined mesh) the hp-refinement $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes possess almost the same convergence rates. For $p_f = 0, 1$ the convergence orders is limited to 1.27 and it is not a surprise because the polynomial degrees p_c have been used in only 11% of triangles of the meshes. For $p_f = 2$ the convergence order is limited to 2.78 despite the polynomial degrees p_c have been used in only 4% of the triangles of the meshes. One can observe that $\forall p_f$, if $p_c = p_f + 1$, the convergence rate is faster than the other cases (e.g. for $p_c = p_f + 2$). These results are in conformity with the formula (6.17) although it seems that this formula is sub-optimal since we have demonstrated here a convergence rate as high as 2.78 suggesting that theoretical convergence study conducted in [19] might be improved in view of the development of h- and hp-adaptive DGTD methods.

$\sqrt{\text{DOF}}$	L^2 error	order	CPU (s)	# time steps	$\sqrt{\text{DOF}}$	L^2 error	order	CPU (s)	# time steps
	$\mathbb{P}_1$:$\mathbb{P}_0$-DG	($\%\text{DOF}_c = 27\%$)				$\mathbb{P}_2$:$\mathbb{P}_0$-DG	($\%\text{DOF}_c = 43\%$)		
26.5	4.10E−02	–	< 1	268	30	4.83E−03	–	< 1	469
53	1.56E−02	1.39	2	×2	60	2.79E−03	0.79	5	×2
106	6.56E−03	1.25	26	×4	120	9.76E−04	1.52	52	×4
212	2.53E−03	1.37	212	×8	240	5.92E−04	0.72	438	×8
424	1.25E−03	1.02	1732	×16					
	$\mathbb{P}_3$:$\mathbb{P}_0$-DG	($\%\text{DOF}_c = 55\%$)				$\mathbb{P}_4$:$\mathbb{P}_0$-DG	($\%\text{DOF}_c = 65\%$)		
34	4.18E−03	–	1	625	38	4.85E−03	–	3	1043
68	2.24E−03	0.90	9	×2	76	2.10E−03	1.21	23	×2
136	1.04E−03	1.11	92	×4	152	1.01E−03	1.05	209	×4
272	5.68E−04	0.87	744	×8	304	5.24E−04	0.95	1768	×8
	$\mathbb{P}_2$:$\mathbb{P}_1$-DG	($\%\text{DOF}_c = 20\%$)				$\mathbb{P}_3$:$\mathbb{P}_1$-DG	($\%\text{DOF}_c = 29\%$)		
44	2.78E−03	–	2	625	46.5	1.52E−03	–	3	750
88	8.83E−04	1.65	13	×2	93	7.49E−04	1.02	19	×2
176	3.93E−04	1.17	130	×4	186	3.71E−04	1.01	180	×4
352	1.93E−04	1.02	1083	×8	372	1.86E−04	0.99	1480	×8
	$\mathbb{P}_4$:$\mathbb{P}_1$-DG	($\%\text{DOF}_c = 38\%$)				$\mathbb{P}_3$:$\mathbb{P}_2$-DG	($\%\text{DOF}_c = 7\%$)		
50	1.49E−03	–	4	1042	57.5	6.94E−04	–	4	800
100	7.35E−04	1.02	34	×2	115	1.04E−04	2.74	32	×2
200	3.69E−04	0.99	311	×4	230	1.49E−05	2.80	267	×4
400	1.86E−04	0.99	2494	×8	460	2.13E−06	2.81	2277	×8
	$\mathbb{P}_4$:$\mathbb{P}_2$-DG	($\%\text{DOF}_c = 10\%$)			Percentage of triangles in the coarse mesh $\%\Delta_c$				
58.5	1.02E−04	–	7	1200					
117	1.66E−05	2.62	51	×2	For $p_f = 0, 1,\ \forall p_c,\ \%\Delta_c$=11%				
234	3.90E−06	2.09	431	×4	For $p_f = 2,\ \forall p_c,\ \%\Delta_c$=4%				
468	9.40E−07	2.05	3665	×8					

Global convergence order of the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes

$\mathbb{P}_1$:$\mathbb{P}_0$	$\mathbb{P}_2$:$\mathbb{P}_0$	$\mathbb{P}_3$:$\mathbb{P}_0$	$\mathbb{P}_4$:$\mathbb{P}_0$	$\mathbb{P}_2$:$\mathbb{P}_1$	$\mathbb{P}_3$:$\mathbb{P}_1$	$\mathbb{P}_4$:$\mathbb{P}_1$	$\mathbb{P}_3$:$\mathbb{P}_2$	$\mathbb{P}_4$:$\mathbb{P}_2$
1.27	1.06	0.98	1.07	1.27	1.01	1.0	2.78	2.24

Table 15: Convergence study for the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes.
Locally refined non-conforming meshes.
$\%\text{DOF}_c$ represents the percentage of DOF in the coarse mesh.
L^2 errors, CPU times and time steps numbers are measured after two periods.

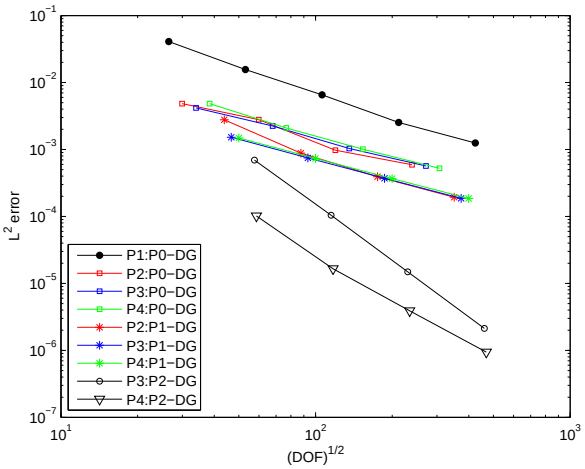


Figure 6.27: Convergence of the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DG schemes. Locally refined non-conforming meshes.

We show on Fig. 6.28 (right) the L^2 error as a function of the CPU time. It is clear that the gains in CPU time are notable if we increase the accuracy of the schemes in space. Fig. 6.28 (left) shows the efficiency curves (the CPU time as a function of $\sqrt{\text{DOF}}$). The corresponding global efficiency orders are given in Tab. 16.

$\mathbb{P}_1$:$\mathbb{P}_0$	$\mathbb{P}_2$:$\mathbb{P}_0$	$\mathbb{P}_3$:$\mathbb{P}_0$	$\mathbb{P}_4$:$\mathbb{P}_0$	$\mathbb{P}_2$:$\mathbb{P}_1$	$\mathbb{P}_3$:$\mathbb{P}_1$	$\mathbb{P}_4$:$\mathbb{P}_1$	$\mathbb{P}_3$:$\mathbb{P}_2$	$\mathbb{P}_4$:$\mathbb{P}_2$
3.16	3.19	3.11	3.08	3.15	3.12	3.10	3.24	3.05

Table 16: Global efficiency orders for the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes.

Finally, Tab. 15 calls for the following remarks:

- the *hp*-like $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method is more accurate than the *p*-enrichment and *h*-refinement DGTD methods $\forall p_c, p_f$. Moreover, to reach a given accuracy, the required number of DOF is less than the one required by the *p*-enrichment and *h*-refinement methods.
- the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method is less expensive in terms of memory storage and CPU time. For example, the CPU time required to obtain accuracies of 4.83E−03 and 5.25E−03 using respectively the $\mathbb{P}_2$:$\mathbb{P}_0$-DGTD and $\mathbb{P}_0$-DGTD schemes (with conforming meshes, see Tab. 12) are < 1 seconds and 133 seconds, likewise for the $\mathbb{P}_2$:$\mathbb{P}_1$-DGTD by comparison with $\mathbb{P}_1$-DGTD, and $\mathbb{P}_3$:$\mathbb{P}_2$-DGTD by comparison with $\mathbb{P}_2$-DGTD and $\mathbb{P}_3$-DGTD.

7 Concluding remarks and Future work

In this work, we have proposed a *hp*-like discontinuous Galerkin $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method for the discretization of the time-domain Maxwell's equations on both conforming and non-conforming locally refined meshes. This method is based on centered numerical fluxes and a leap-frog time-scheme, and the unknowns are approximated with discontinuous nodal polynomials of degree that

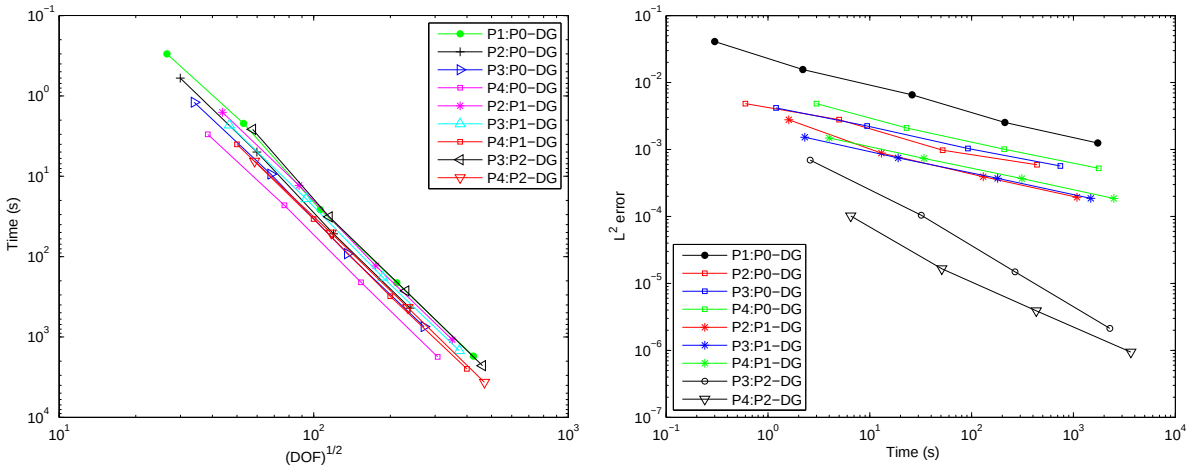


Figure 6.28: CPU time as a function of $\sqrt{\text{DOF}}$ (left) and L^2 error as a function of the CPU time (right). $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD schemes using locally refined non-conforming meshes.

may vary over different elements of the mesh. The $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method conserves a discrete energy and is stable under a CFL condition. It can be seen like an extension of the $\mathbb{P}_p$-DGTD formulation recently studied in [18]. It is presented in this work with a new implementation which allows h-refinement and p-enrichment and which is based on a numerical quadrature formula for the computation of the flux matrices associated to non-conforming interfaces. Finally, numerical experiments show that the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method can reduce (or eliminate) the numerical dispersion errors resulting from the $\mathbb{P}_p$-DGTD ($p = 0, 1$) method, and these errors do not increase when using small time steps. Numerical convergence study are presented for p- h- and hp-refinement methods using local and global refinement. We can conclude that the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method is the most accurate, the less expensive in terms of memory storage and CPU time, and requires a number of degrees of freedom smaller than the one required by the h-refinement and p-enrichment methods.

This work is a first step toward the development of an hp-adaptive method for Maxwell's equations. The continuation of this study relates to various facets of the $\mathbb{P}_{p_c}$:$\mathbb{P}_{p_f}$-DGTD method proposed here:

- the validation of this method on test cases involving complex geometries and heterogeneous materials,
- further theoretical studies of the properties of the method,
- extension to the three-dimensional case.

Acknowledgment. The authors are indebted to Loula Fezoui for valuable discussions on this work.

References

[1] M. Bernacki, L. Fezoui, S. Lanteri, and S. Piperno. Parallel unstructured mesh solvers for heterogeneous wave propagation problems. *Appl. Math. Model.*, 30(8):744–763, 2006.

[2] A. Bossavit. A rationale for edge-elements in 3-D fields computations. *IEEE Trans. Magnetics*, 24:74–79, 1988.

[3] A. Bossavit. Solving Maxwell equations in a closed cavity, and the question of spurious modes. *IEEE Trans. Magnetics*, 26:702–705, 1990.

[4] A. Buffa, P. Houston, and I. Perugia. Discontinuous Galerkin computation of the Maxwell eigenvalues on simplicial meshes. *J. Comput. Appl. Math.*, 2007. To appear.

[5] N. Canouet. *Méthodes de Galerkin discontinu pour la résolution du système de Maxwell sur des maillages localement raffinés non-conformes.* PhD thesis, Ecole Nationale des Ponts et Chaussées, 2003.

[6] N. Canouet, L. Fezoui, and S. Piperno. Discontinuous Galerkin time-domain solution of Maxwell's equations on locally-refined nonconforming cartesian grids. *COMPEL*, 24(4):1381–1401, 2005.

[7] W. Cecot, W. Rachowicz, and L. Demkowicz. An *hp*-adaptive finite element method for electromagnetics. Part 3: a three-dimensional infinite element for Maxwell's equations. *Int. J. Numer. Meth. Engng*, 57:899–921, 2003.

[8] M.-H. Chen, B. Cockburn, and F. Reitich. High-order RKDG methods for computational electromagnetics. *J. Sci. Comput.*, 22:205–226, 2005.

[9] J-P. Cioni, L. Fezoui, L. Anne, and F. Poupaud. A parallel FVTD Maxwell solver using 3D unstructured meshes. In *13th Annual Review of Progress in Applied Computational Electromagnetics*, pages 359–365, Monterey, CA, USA, 1997.

[10] B. Cockburn, G.E. Karniadakis, and C.W. Shu, editors. *Discontinuous Galerkin methods. Theory, computation and applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 2000.

[11] B. Cockburn, F. Li, and C.-W. Shu. Locally divergence-free discontinuous Galerkin methods for the Maxwell equations. *J. Comput. Phys.*, 194:588–610, 2004.

[12] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin method for conservation laws II: General framework. *Math. Comp.*, 52(186):411–435, 1989.

[13] F. Collino, T. Fouquet, and P. Joly. Conservative space-time mesh refinement methods for the FDTD solution of Maxwell's equations. *J. Comp. Phys.*, 211(1):9–35, 2006.

[14] L. Demkowicz and W. Rachowicz. A 2D *hp*-adaptive finite element package for electromagnetics. Technical Report 97-18, TICAM, 1997.

[15] L. Demkowicz and L. Vardapetyan. Modeling of electromagnetic absorption/scattering problems using *hp*-adaptive finite elements. *Computer Meth. Appl. Mech. Engng.*, 152:103–124, 1998.

[16] G. Dhatt and G. Touzot. *Une présentation de la méthode des éléments finis.* Collection Université de Compiègne. Maloine S.A. Éditeur, Paris, second edition, 1984.

[17] K. Dridi, J. S. Hesthaven, and A. Ditkowski. Staircase-free finite-difference time-domain formulation for general materials in complex geometries. *IEEE Trans. Antennas Propag.*, 49:749–755, 2001.

[18] H. Fahs, S. Lanteri, and F. Rapetti. Etude de stabilité d'une méthode Galerkin discontinu pour la résolution numérique des équations de Maxwell 2D en domaine temporel sur des maillages triangulaires non-conformes. Research Report 6023, INRIA, Juillet 2006.

[19] L. Fezoui, S. Lanteri, S. Lohrengel, and S. Piperno. Convergence and stability of a discontinuous Galerkin time-domain method for the heterogeneous Maxwell equations on unstructured meshes. *ESAIM: Math. Model. and Numer. Anal.*, 39(6):1149–1176, 2006.

[20] M. Fusco. FDTD algorithm in curvilinear coordinates. *IEEE Trans. Antennas Propagat.*, 38:76–89, 1990.

[21] J. S. Hesthaven. High-order accurate methods in time-domain computational electromagnetics. *Adv. Imaging Elec. Phys.*, 127:59–123, 2003.

[22] J.S. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids. I. Time-domain solution of Maxwell's equations. *J. Comput. Phys.*, 181(1):186–221, 2002.

[23] J.S. Hesthaven and T. Warburton. High-order accurate methods for time-domain electromagnetics. *Comp. Mod. Engin. Sci*, 5(5):395–408, 2004.

[24] J.S. Hesthaven and T. Warburton. High-order nodal discontinuous Galerkin methods for the Maxwell's eigenvalue problem. *Royal Soc. London Ser. A*, 362:493–524, 2004.

[25] R. Holland. Finite difference solutions of Maxwell's equations in generalized nonorthogonal coordinates. *IEEE Trans. Nuclear Sci.*, 30:4589–4591, 1983.

[26] P. Houston, R. Hartmann, and E. Süli. Adaptive discontinuous Galerkin finite element methods for compressible fluid flows. In M. Baines, editor, *Numerical Methods for Fluid Dynamics VII*, pages 347–353, 2001.

[27] P. Houston, I. Perugia, and D. Schötzau. *hp*-DGFEM for Maxwell's equations. In F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, editors, *Numerical Mathematics and Advanced Applications (ENUMATH 2001)*, pages 785–794. Springer-Verlag, 2003.

[28] P. Houston, I. Perugia, and D. Schötzau. Mixed discontinuous Galerkin approximation of the Maxwell operator. *SIAM J. Numer. Anal*, 42(1):434–459, 2004.

[29] P. Houston, I. Perugia, and D. Schötzau. Mixed discontinuous Galerkin approximation of the Maxwell operator: non-stabilized formulation. *J. Sci. Comp.*, 22:325–356, 2005.

[30] P. Houston, B. Senior, and E. Süli. *hp*-discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity. *Internat. J. Numer. Meth. Fluids*, 40(1–2):153–169, 2002.

[31] T. G. Jurgens, A. Taflove, K. Umashankar, and T.G. Moore. Finite-difference time-domain modeling of curved surfaces. *IEEE Trans. Antennas Propagat.*, 40:357–366, 1992.

[32] D. Kopriva, S.L. Woodruff, and M.Y. Hussaini. Discontinuous spectral element approximation of Maxwell's equations. In B. Cockburn, G.E. Karniadakis, and C.W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 355–362. Springer-Verlag, 2000.

[33] D. Kopriva, S.L. Woodruff, and M.Y. Hussaini. Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. *Int. J. Numer. Meth. Engng.*, 53:105–122, 2002.

[34] A.S. Lebedev, M.P. Fedoruk, and O.V. Shtyrina. Finite-volume algorithm for solving the time-dependent Maxwell equations on unstructured meshes. *Comput. Mathematics and Mathematical Phys.*, 46:1219–1233, 2006.

[35] P. Monk. *Finite element methods for Maxwell's equations*. Oxford University Press, 2003.

[36] J. C. Nedelec. A new family of mixed finite elements in $\mathbb{R}^3$. *Numer. Math.*, 50:57–81, 1986.

[37] J.C. Nedelec. Mixed finite elements in $\mathbb{R}^3$. *Numer. Math.*, 35:315–341, 1980.

[38] D. Pardo, L. Demkowicz, and J. Gopalakrishman. Integration of *hp*-adaptivity and a two grid solver. II. Electromagnetic problems. Technical Report 04-58, ICES, 2004.

[39] K. D. Paulsen and D. R. Lynch. Elimination of vector parasites in finite element Maxwell solutions. *IEEE Trans. Microwave Theory Tech.*, 39:395–404, 1991.

[40] S. Piperno, M. Remaki, and L. Fezoui. A nondiffusive finite volume scheme for the three-dimensional Maxwell's equations on unstructured meshes. *SIAM J. Numer. Anal.*, 39(6):2089–2108, 2002.

[41] A.C. Polycarpou, C.A. Balanis, and A. Stefanov. Helicopter rotor-blade modulation of antenna radiation characteristics. *IEEE Trans. Antennas Propagat.*, 49:688–696, 2001.

[42] W. Rachowicz and L. Demkowicz. An *hp*-adaptive finite element method for electromagnetics. Part 2: A 3D implementation. *Int. J. Numer. Meth. Engng.*, 53:147–180, 2002.

[43] W. Rachowicz and A. Zdunek. An *hp*-adaptive finite element method for scattering problems in computational electromagnetics. *Int. J. Numer. Engng.*, 62:1226–1249, 2005.

[44] F. Reitich and K.K. Tamma. State-of-the-art, trends and directions in computational electromagnetics. *Comput. Meth. Eng. Sci.*, 5:287–294, 2004.

[45] J.-F. Remacle, J. Flaherty, and M. Shephard. An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Review*, 45(1):53–72, 2003.

[46] M. Remaki. A new finite volume scheme for solving Maxwell's system. *COMPEL*, 19(3):913–931, 2000.

[47] G. Scarella, O. Clatz, S. Lanteri, G. Beaume, S. Oudot, J.-P. Pons, S. Piperno, P. Joly, and J. Wiart. Realistic numerical modelling of human head tissue exposure to electromagnetic waves from cellular phones. *Comptes Rendus Physique*, 7(5):501–508, 2006.

[48] P. Solín and J. Cervený. Arbitrary-level hanging nodes and automatic adaptivity in the *hp*-FEM. Research Report 2006-07, University of Texas at El Paso, 2006.

[49] D. Sun, J. Manges, X. Yuan, and Z. Cendes. Spurious modes in finite element methods. *IEEE Antennas Propagat. Mag.*, 37:12–24, 1995.

[50] A. Taflove. *Advances in computational electrodynamics, the finite-difference time-domain method*. Artech House, Boston, London, 1998.

[51] I. Toulopoulos and J.A. Ekaterinaris. High-order discontinuous Galerkin discretizations for computational aeroacoustics in complex domains. *AIAA J.*, 44:502–511, 2006.

[52] L. Vardapetyan and L. Demkowicz. *hp*-adaptive finite elements in electromagnetics. *Comput. Meth. Appl. Mech. Engng.*, 169:331–344, 1999.

[53] R. Verfürth. On the constants in some inverse inequalities for finite element functions. Ruhr-Universität Bochum. unpublished, 1999.

[54] T. Warburton. Application of the discontinuous Galerkin method to Maxwell's equations using unstructured polymorphic *hp*-finite elements. In B. Cockburn, G.E. Karniadakis, and C.W. Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 451–458. Springer-Verlag, 2000.

[55] T. Warburton and J. S. Hesthaven. On the constants in *hp*-finite element trace inequalities. *Comput. Meth. Appl. Mech. Engng.*, 192:2765–2773, 2003.

[56] T. Xiao and Q.H. Liu. A staggered upwind embedded boundary (SUEB) method to eliminate the FDTD staircasing error. *IEEE Trans. Antennas and Propagat.*, 52(3):730–741, 2004.

[57] K.S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas and Propagat.*, 14(3):302–307, 1966.

[58] A. R. Zakharian, M. Brio, and J. V. Moloney. FDTD based second-order accurate local mesh refinement method for Maxwell's equations in two space dimensions. *Comm. Math. Sci.*, 2(3):497–513, 2004.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399